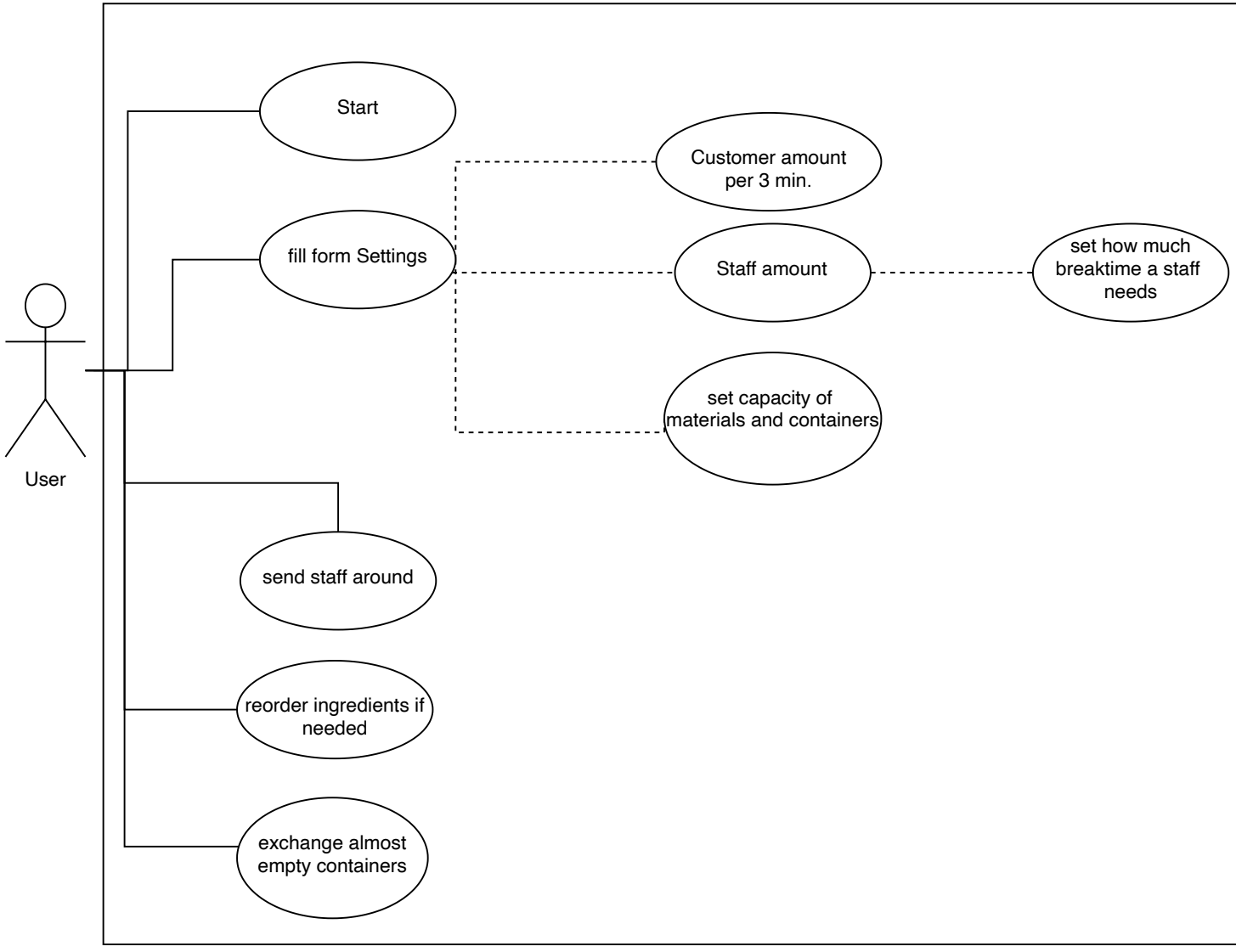
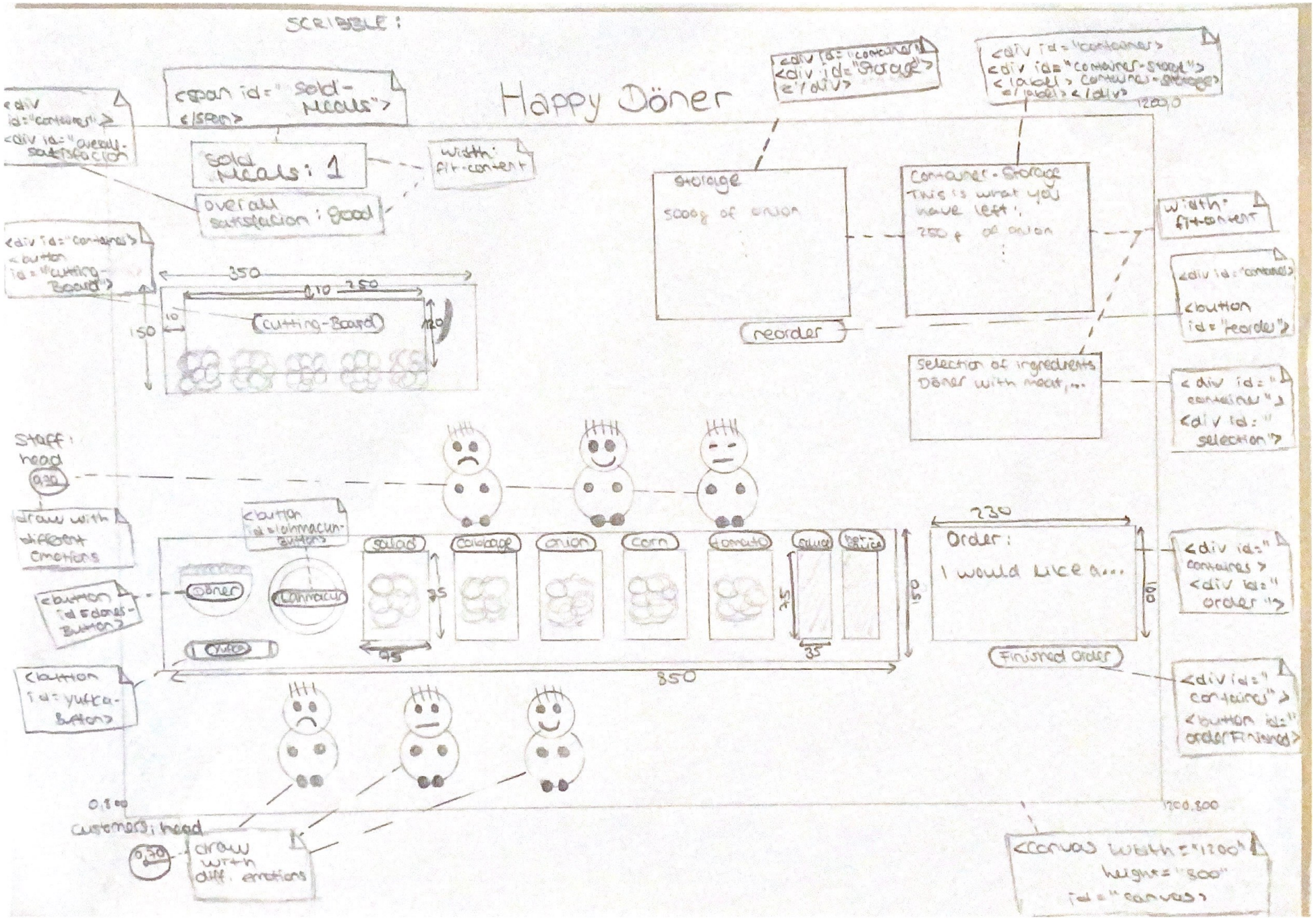
Abschlussarbeit W21

Döner-Trainer

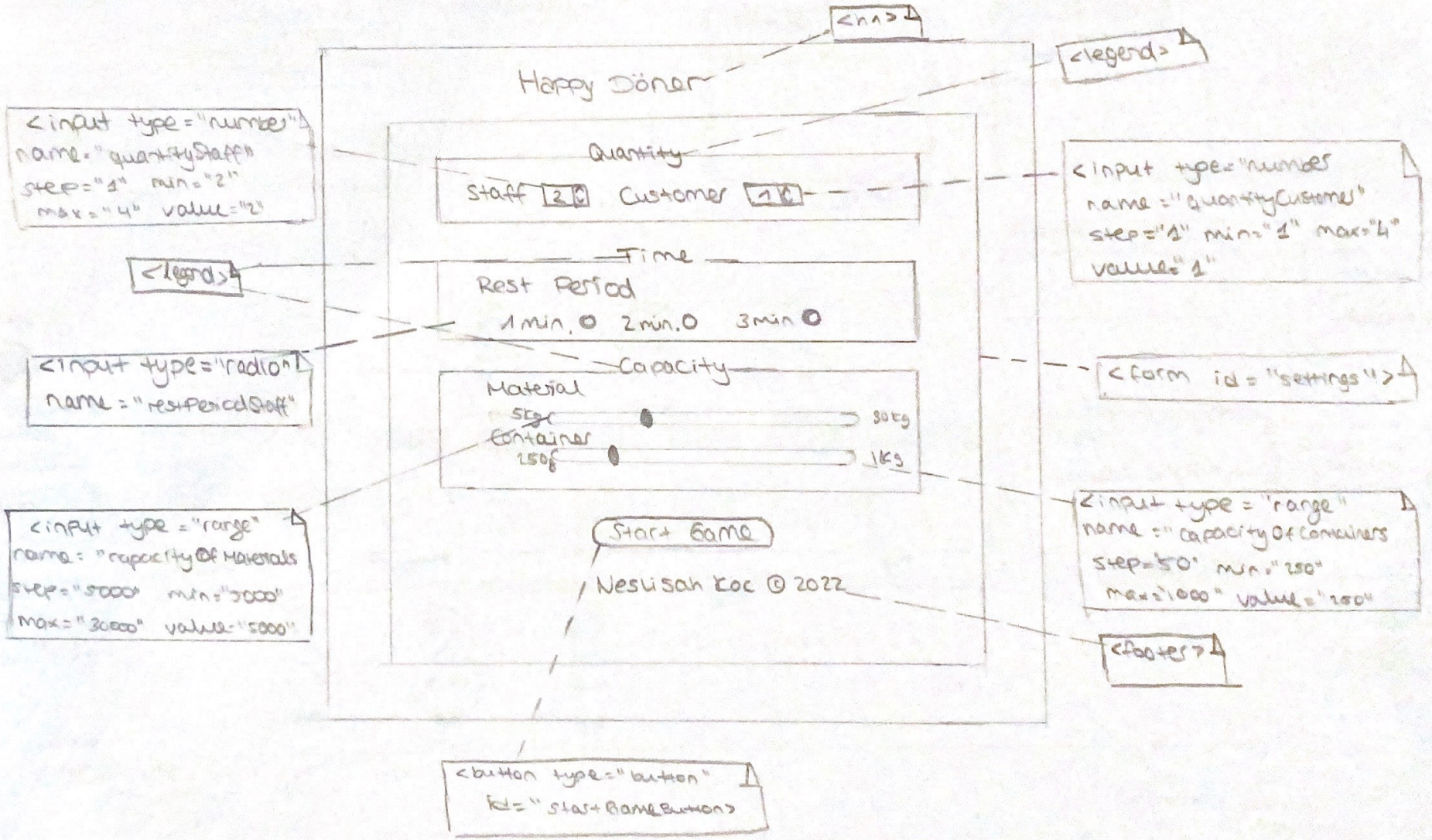
Neslisah Koc / MKB2 / 270155

Zusammenarbeit mit : Verena Rothweiler / MKB2 / 270156

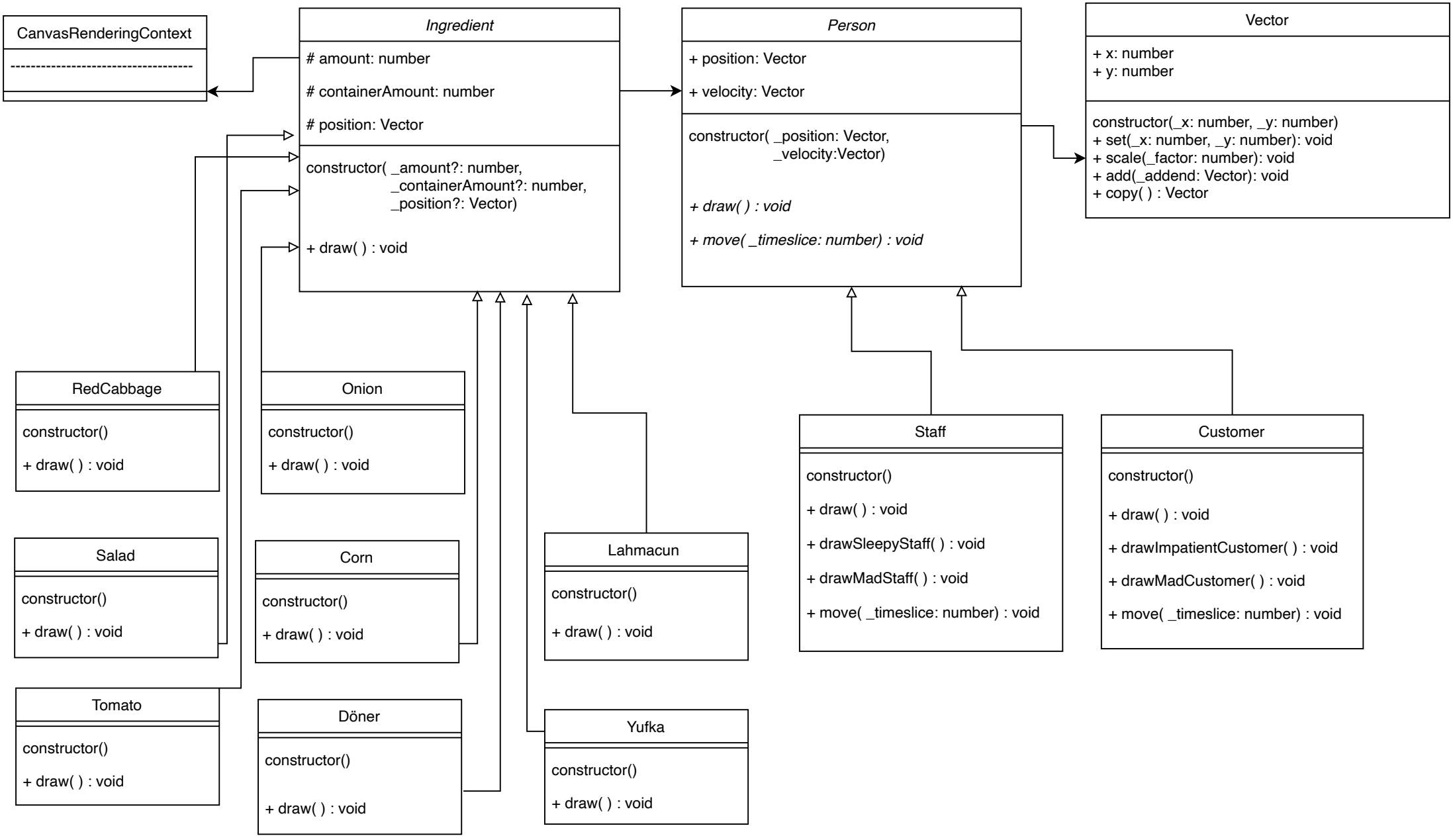



SCRIBBLE:
Happy Döner
<span id="sold-Meals">
</span>
<div id="container">
<div id="overall-satisfaction
Sold Meals: 1
overall satisfaction: good
width: fit-content
<div id="container">
<div id="storage">
</div>
<div id="container">
<div id="container-storage">
</label> container-storage>
</label> </div>
1200,0
storage
5000g of onion
Container-Storage
This is what you have left:
250 g of onion
width: fit-content
<div id="container">
<button id="cutting-Board">
350
10
250
150
10
120
cutting-Board
reorder
<div id="container">
<button id="reorder">
Selection of ingredients
Döner with meat,...
<div id="container">
<div id="selection">
Staff: head
0,30
draw with different emotions
<button id=lahmacun-buttons>
<button id=döner-Button>
Döner
Lahmacun
Yufka
salad
cabbage
onion
corn
tomato
sauce
hot-sauce
75
95
75
35
150
850
230
Order:
I would like a...
100
Finished order
<div id="container">
<div id="order">
<button id=yufka-Button>
<div id="container">
<button id="orderFinished>
0,800
customers: head
0,30
draw with diff. emotions
1200,800
<canvas width="1200"
height="800"
id="canvas">

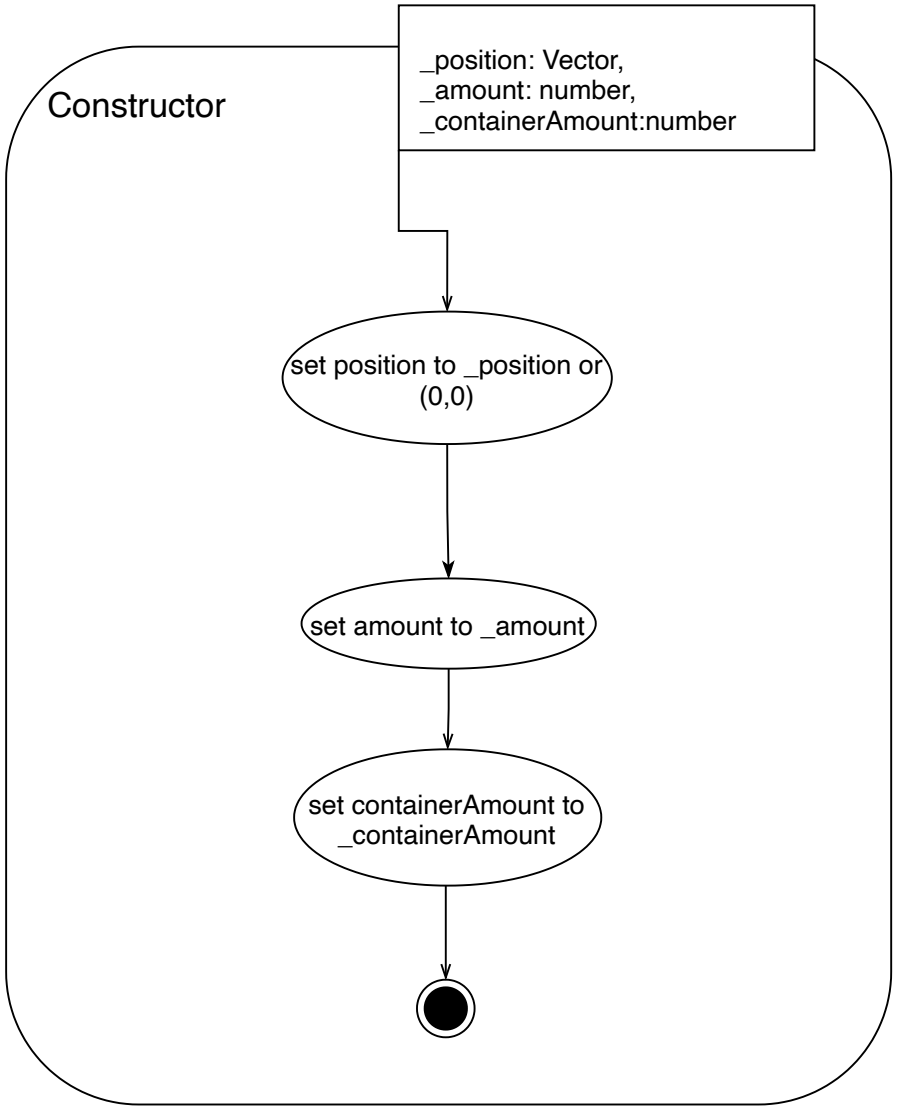
SCRIBBLE:

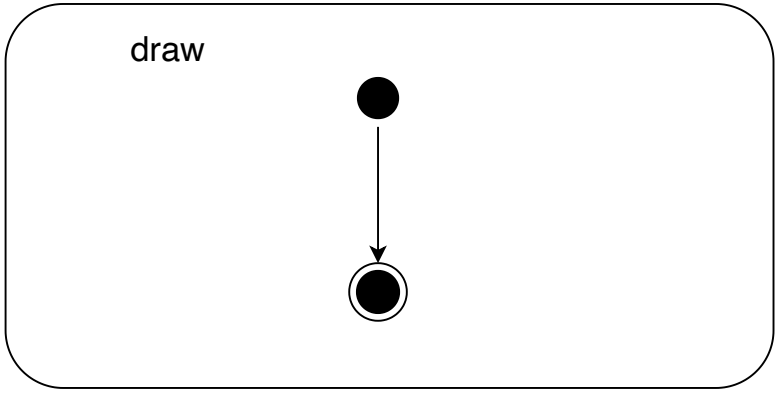

<h1>
Happy Döner
<legend>
Quantity
Staff 2
Customer 1
<input type="number"
name="quantityStaff"
step="1" min="2"
max="4" value="2"
<input type="number
name="quantityCustomer"
step="1" min="1" max="4"
value="1"
<legend>
Time
Rest Period
1 min. 2 min. 3 min
<input type="radio"
name="restPeriodStaff"
Capacity
Material
Steel
Container
250g
30kg
1kg
<form id="settings">
<input type="range"
name="capacityOfMaterials
step="5000" min="5000"
max="30000" value="5000"
Start Game
Neslisah Koc © 2022
<input type="range"
name="capacityOfContainers
step="50" min="250"
max="1000" value="250"
<footer>
<button type="button"
id="startGameButton">

Class Diagram

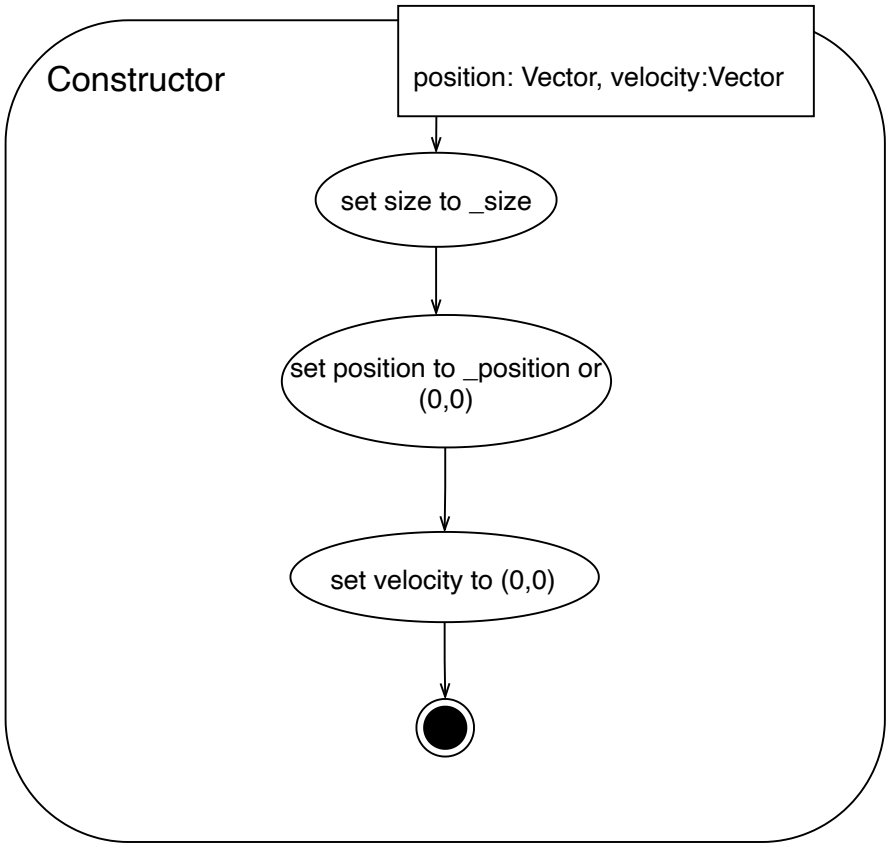


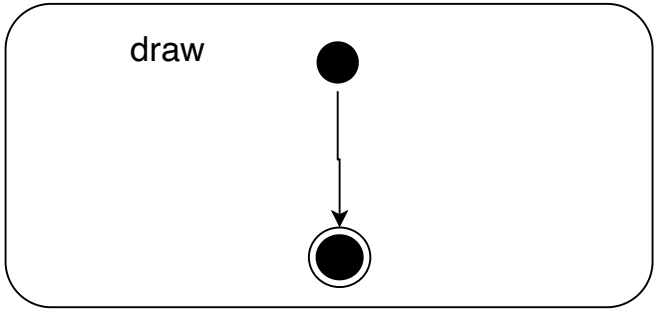
Ingredient: Activity Diagram

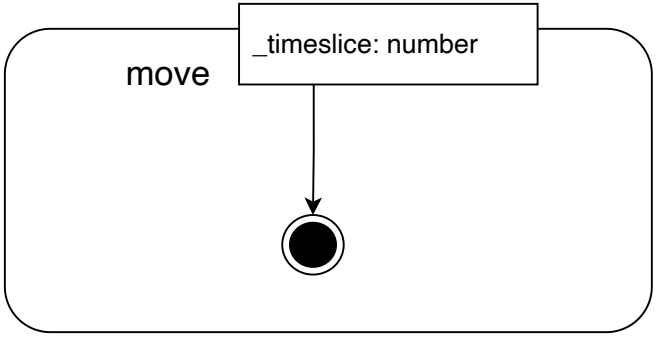




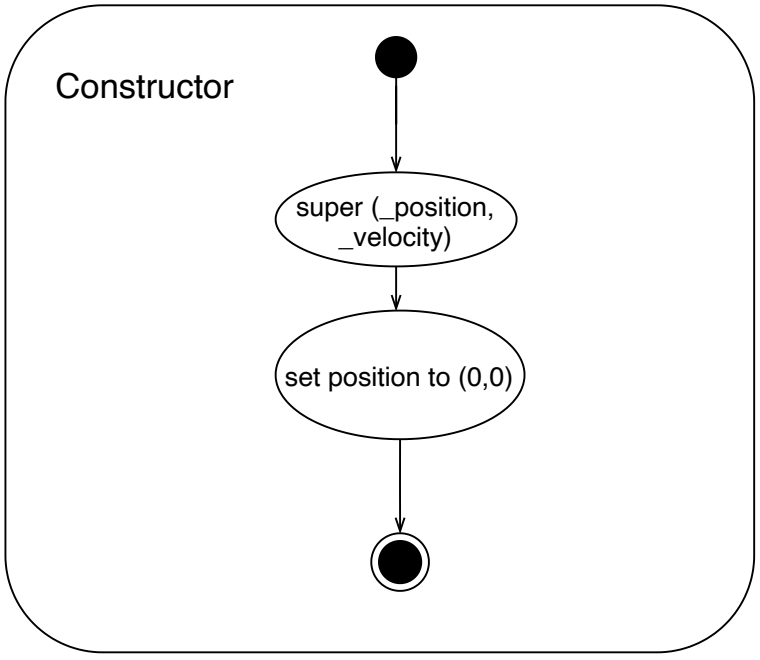
Person: Activity Diagram

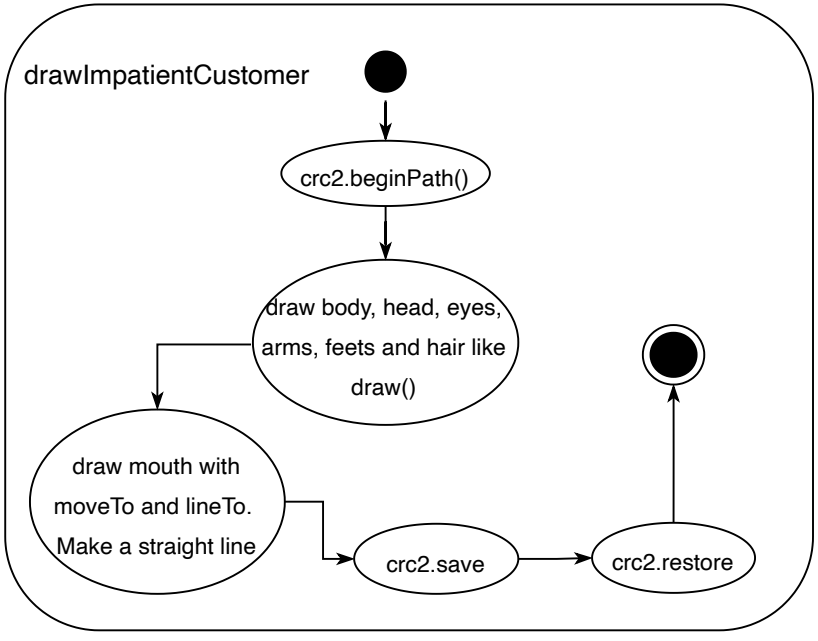


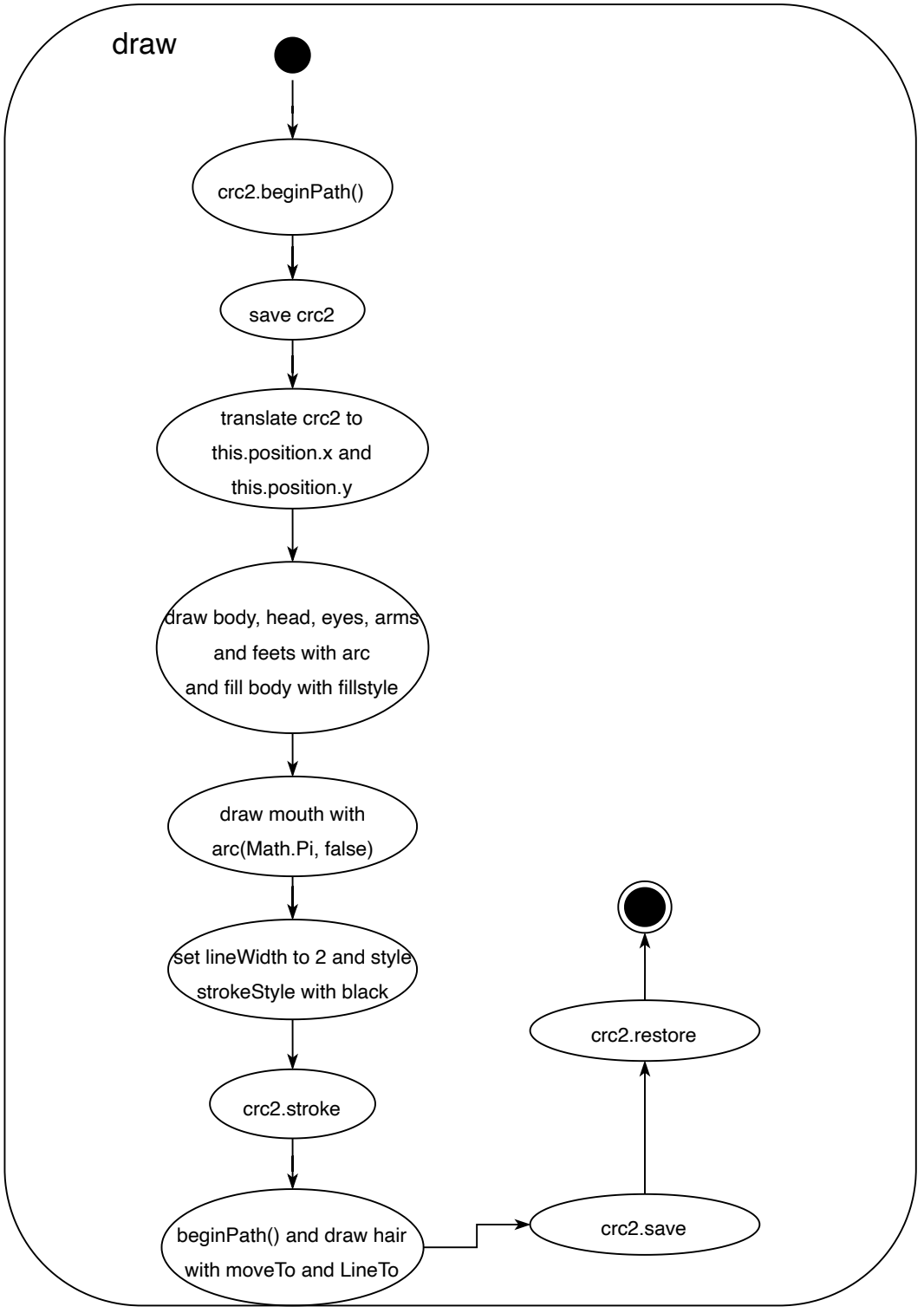


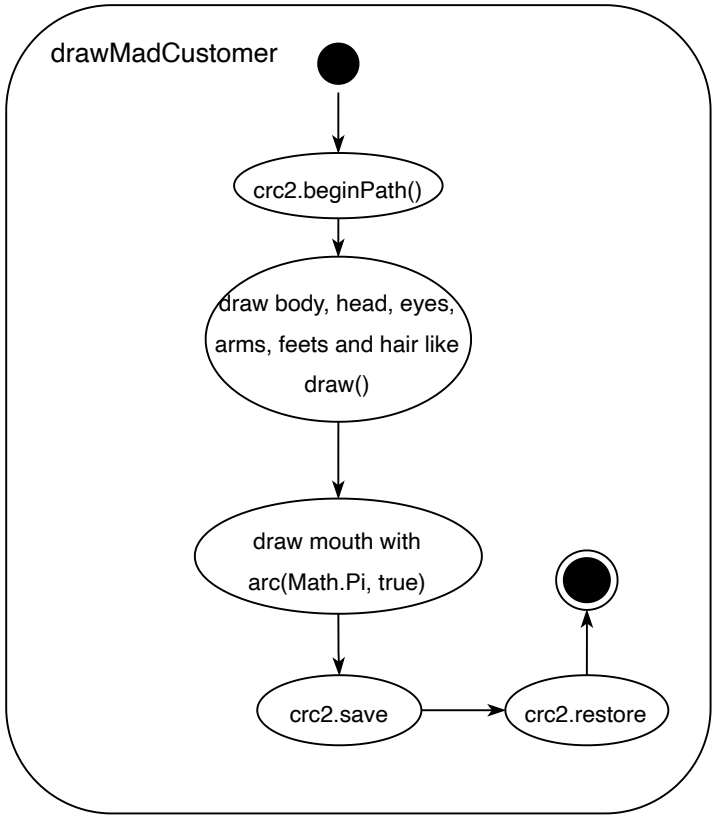


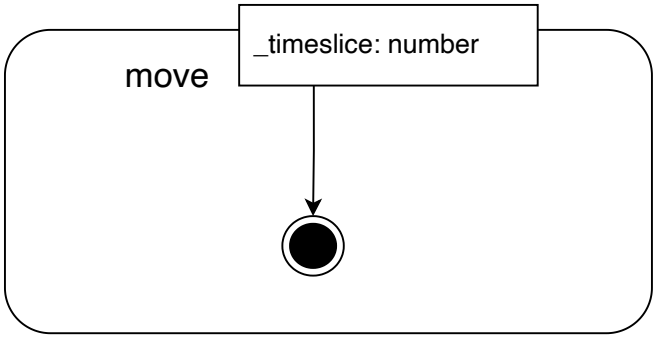
Customer: Activity Diagram



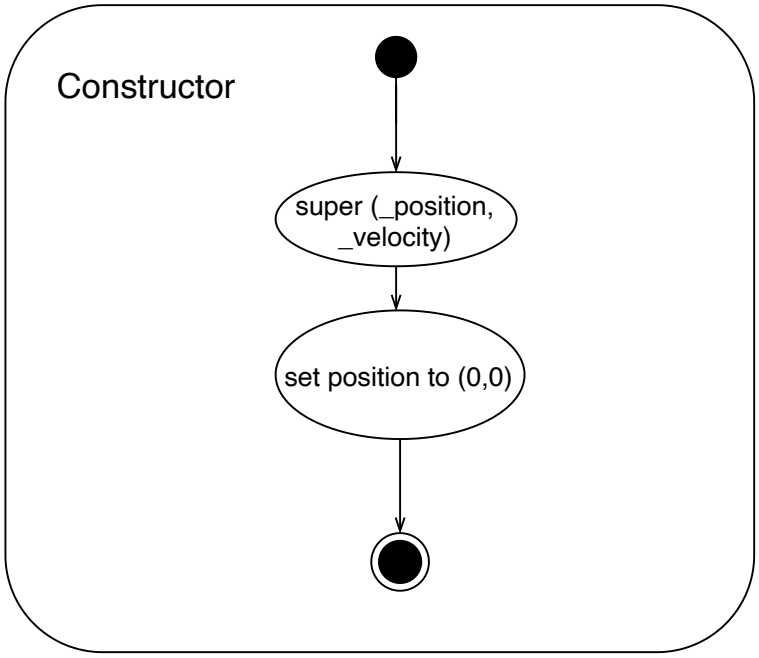


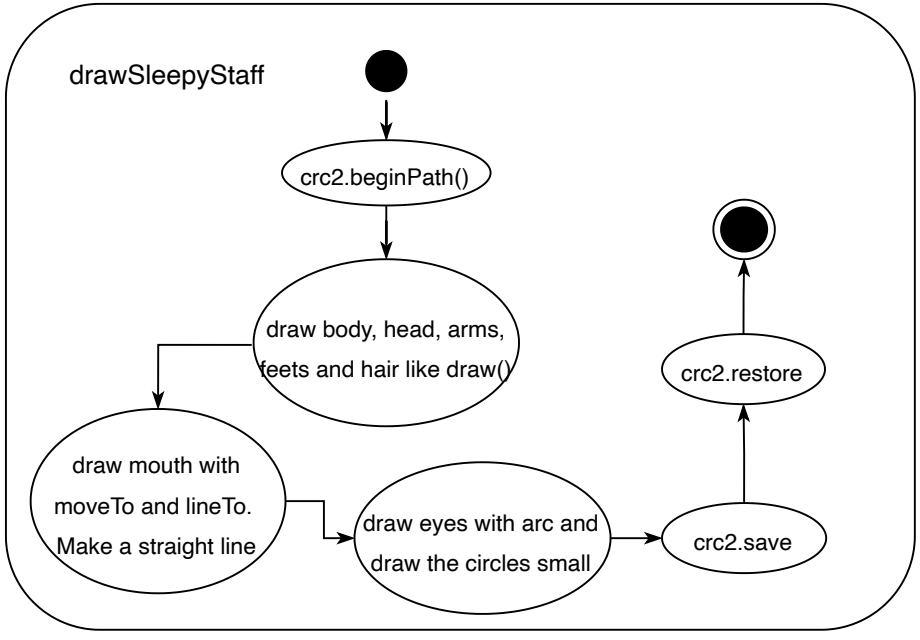


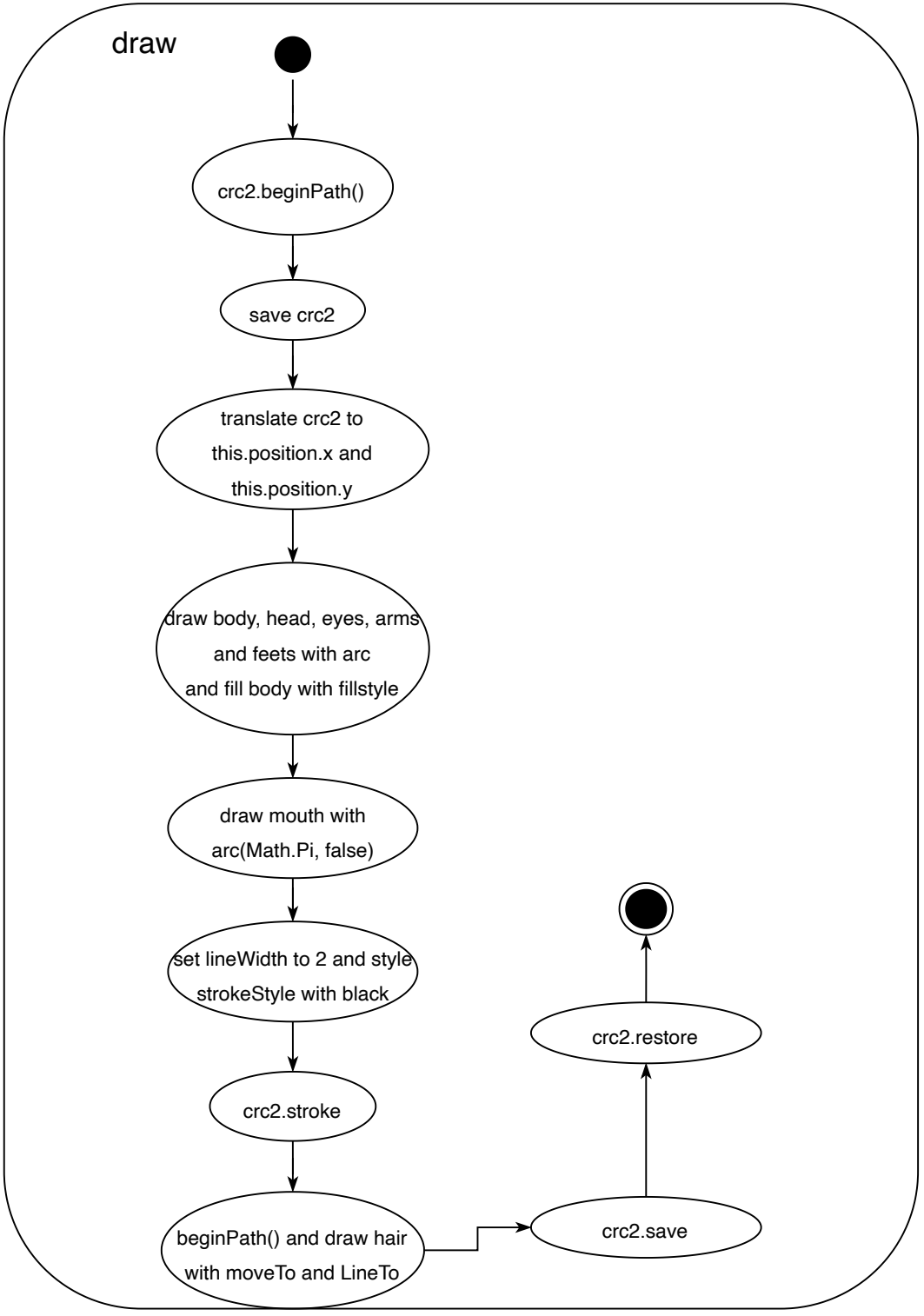

drawMadCustomer
crc2.beginPath()
draw body, head, eyes, arms, feets and hair like draw()
draw mouth with arc(Math.Pi, true)
crc2.save
crc2.restore

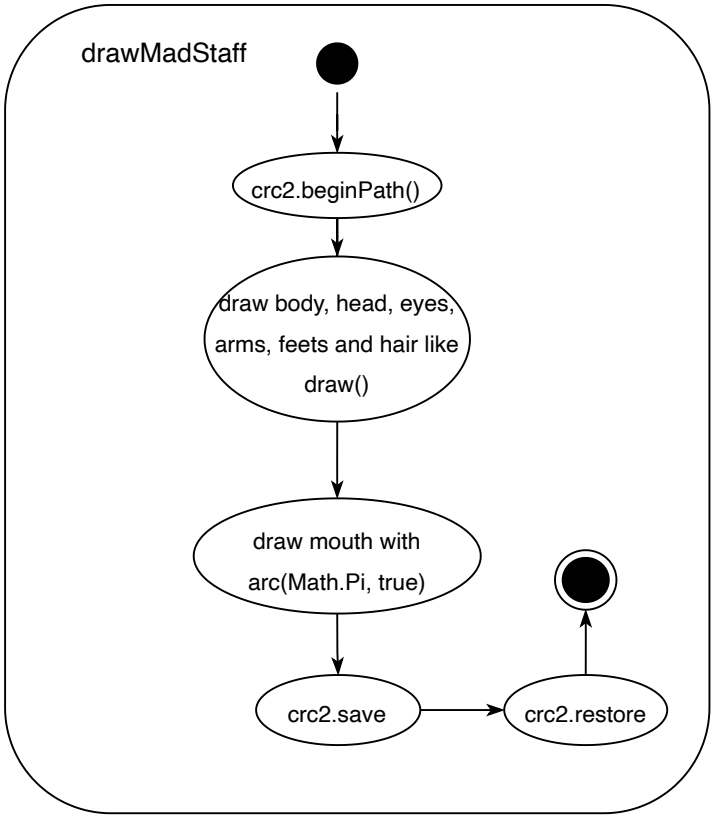

move
_timeslice: number

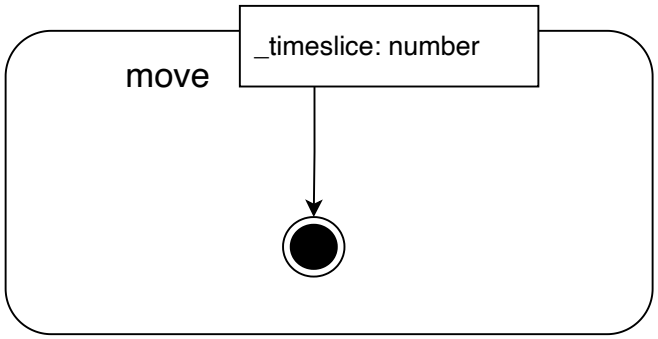
Staff: Activity Diagram



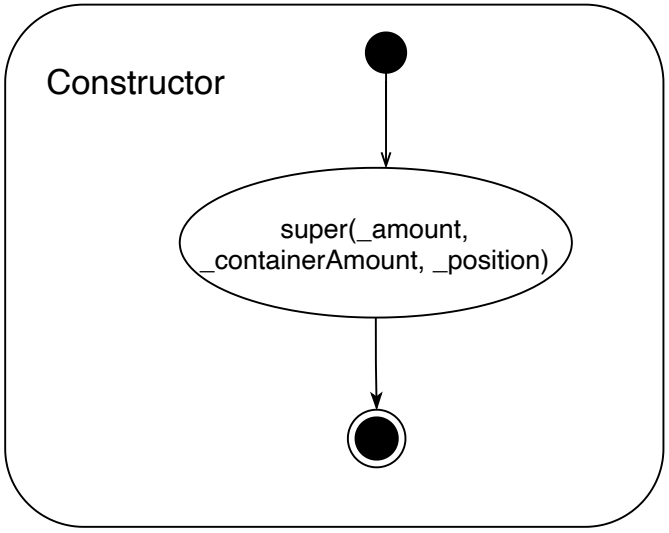


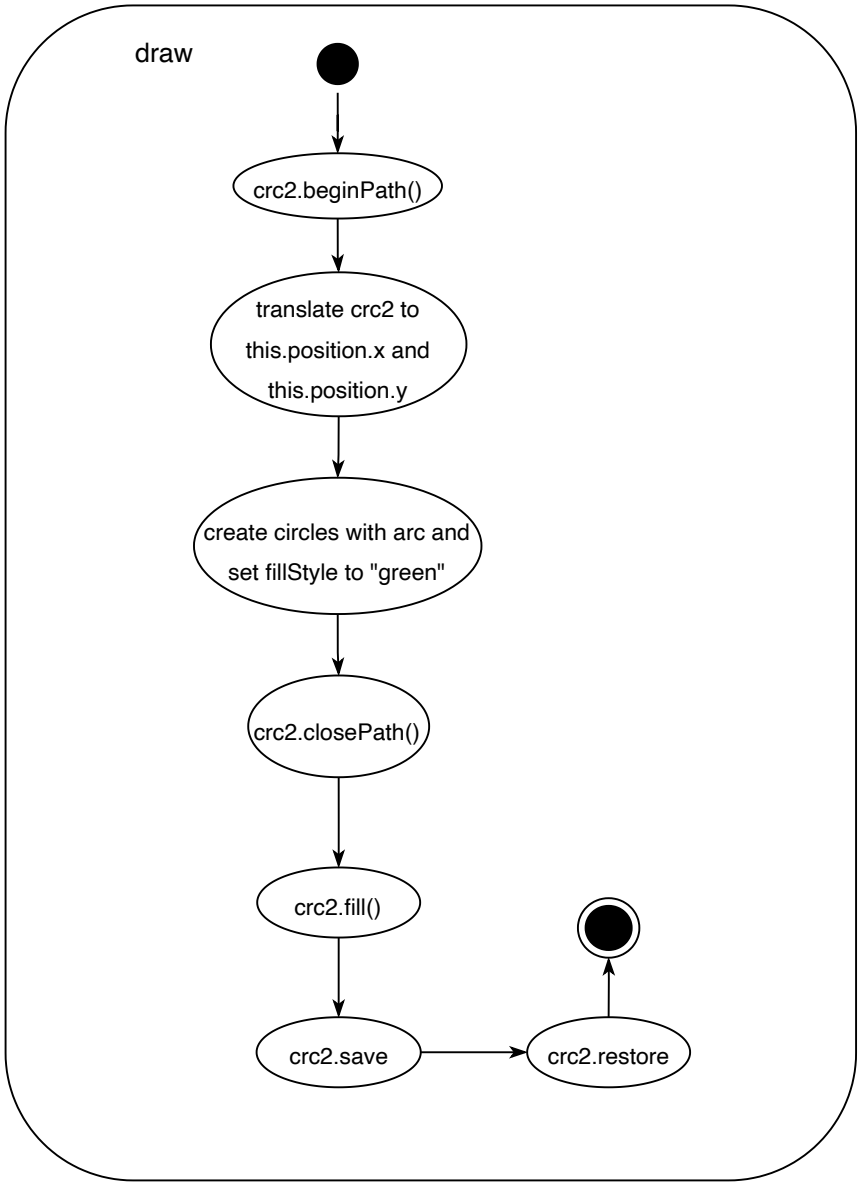



drawMadStaff
crc2.beginPath()
draw body, head, eyes, arms, feets and hair like draw()
draw mouth with arc(Math.Pi, true)
crc2.save
crc2.restore

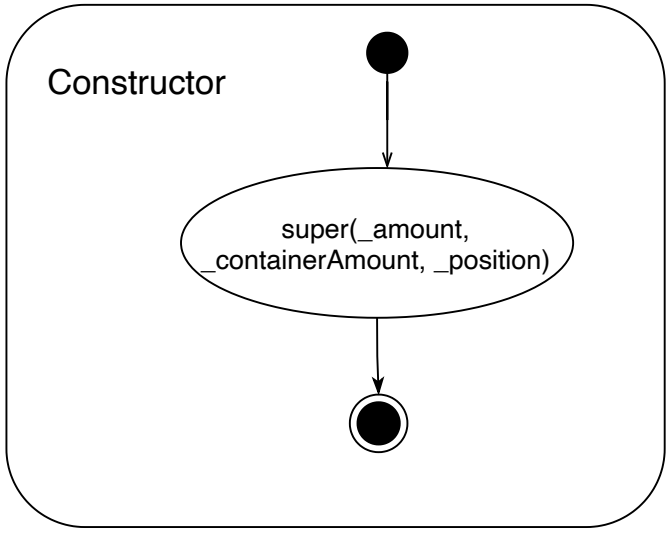

move
_timeslice: number

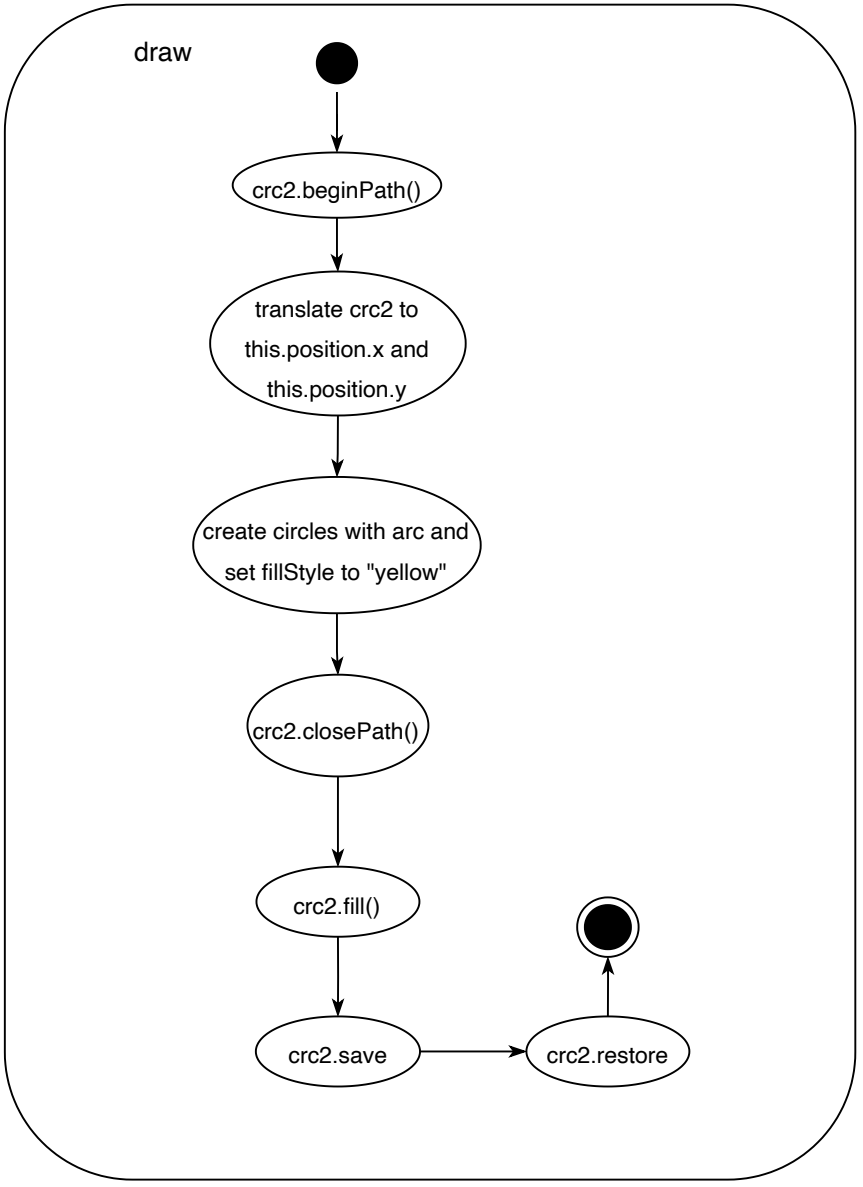
Salad: Activity Diagram



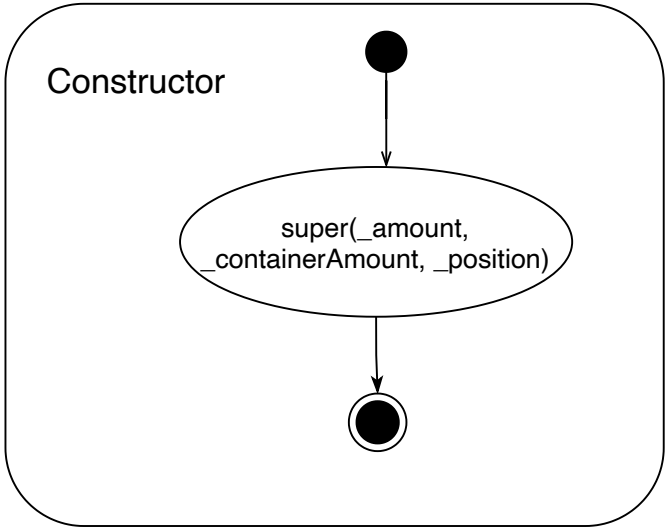


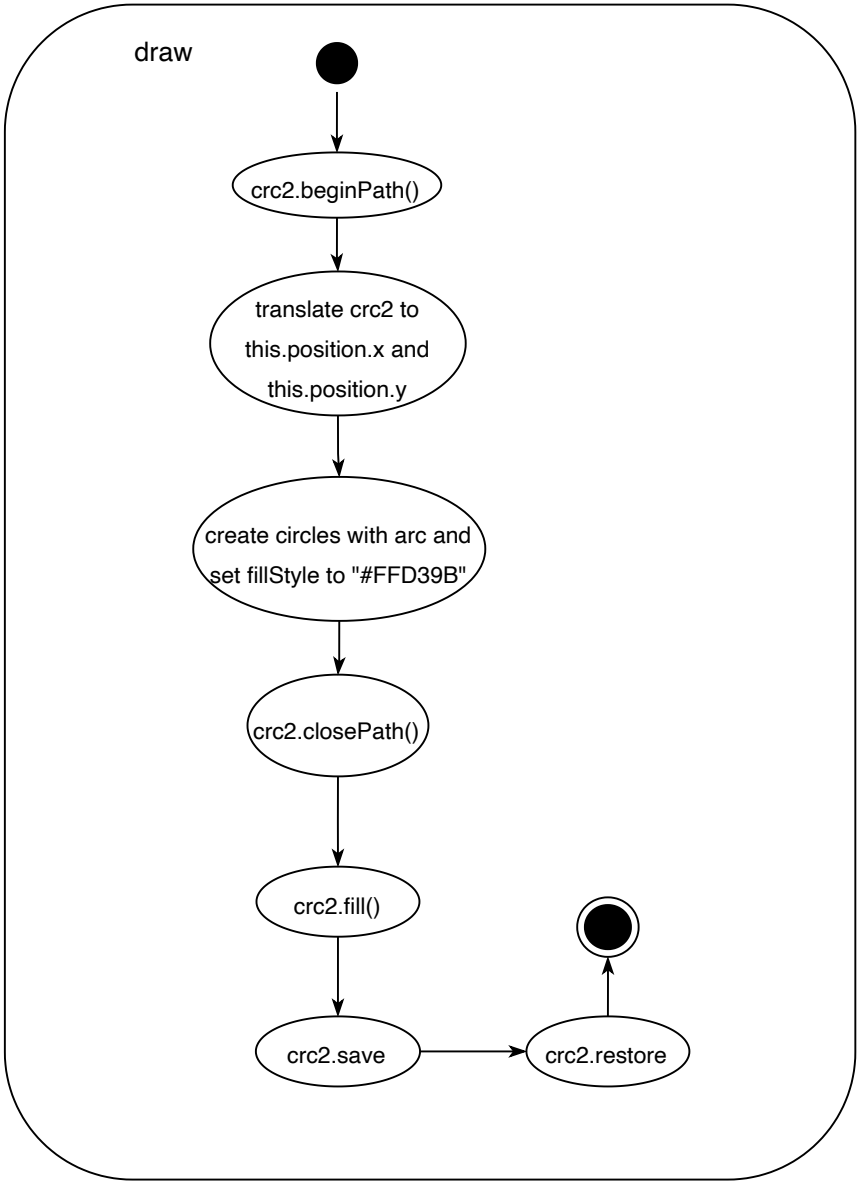
Corn: Activity Diagram



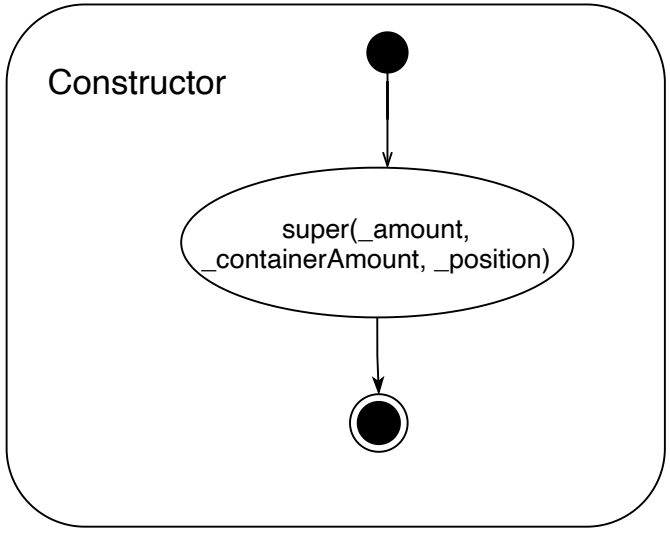


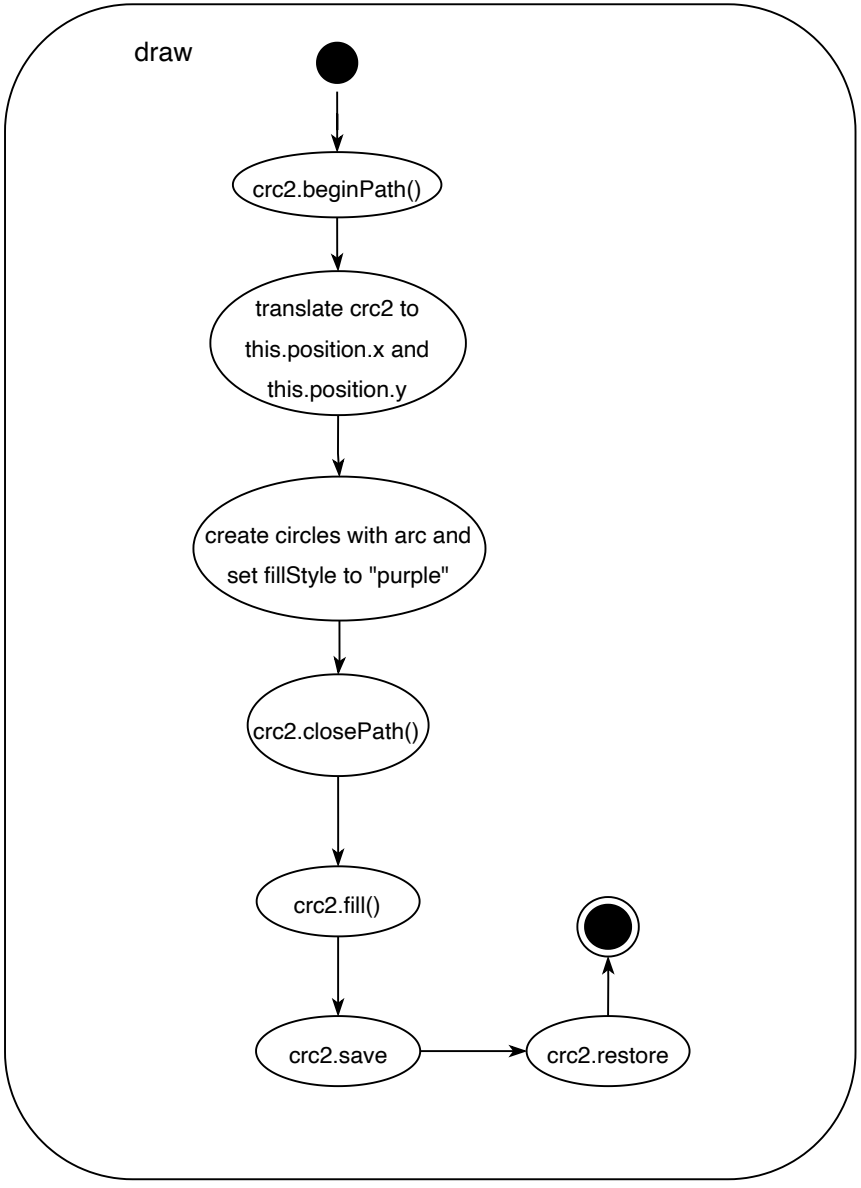
Onion: Activity Diagram



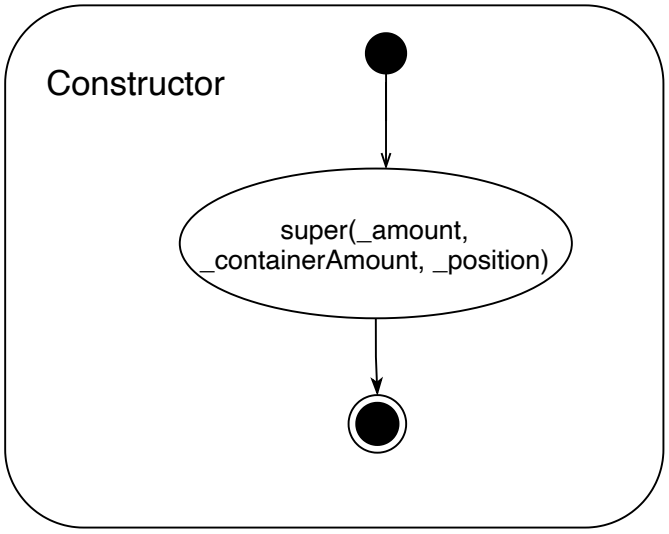


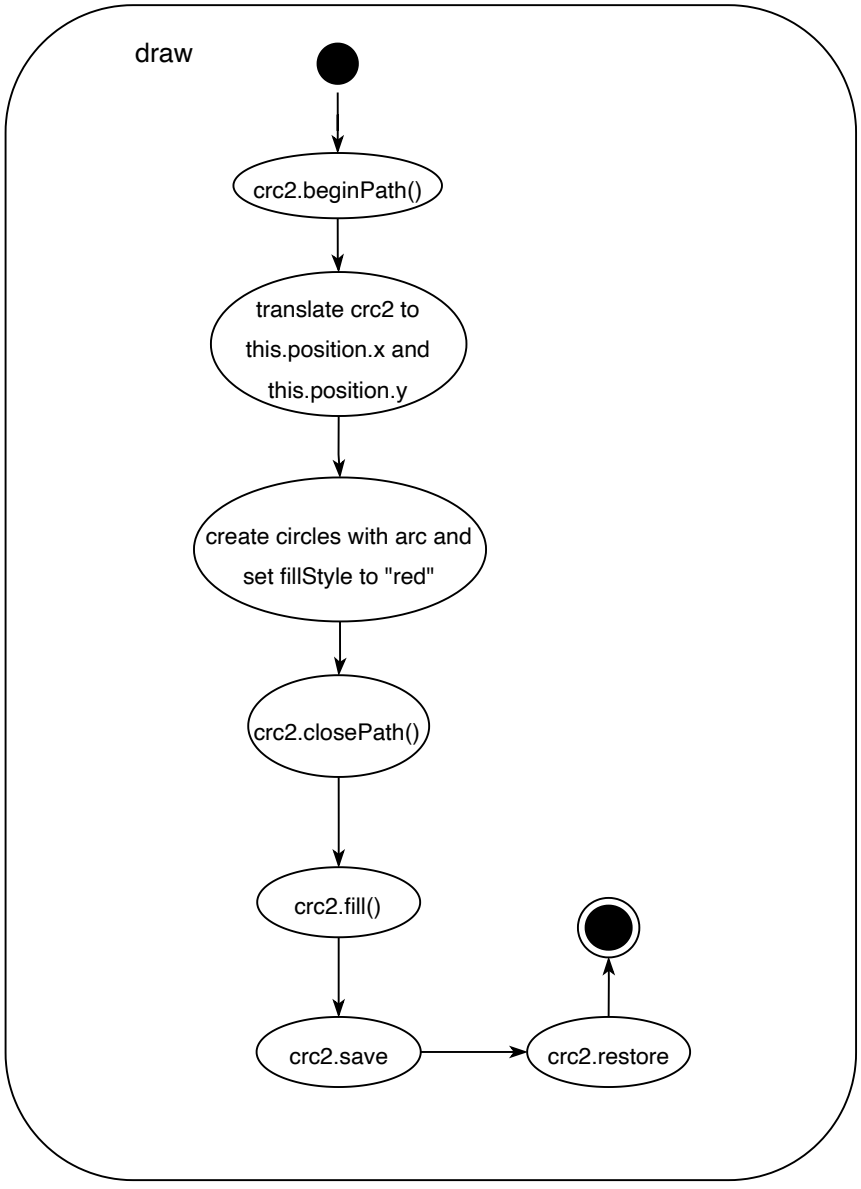
RedCabbage: Activity Diagram



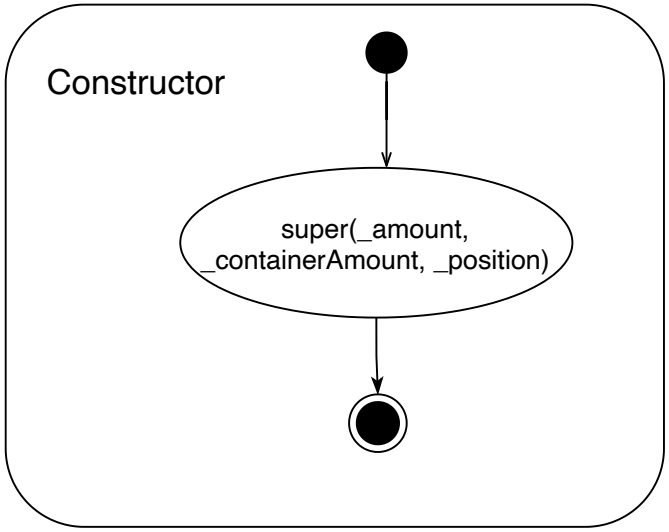


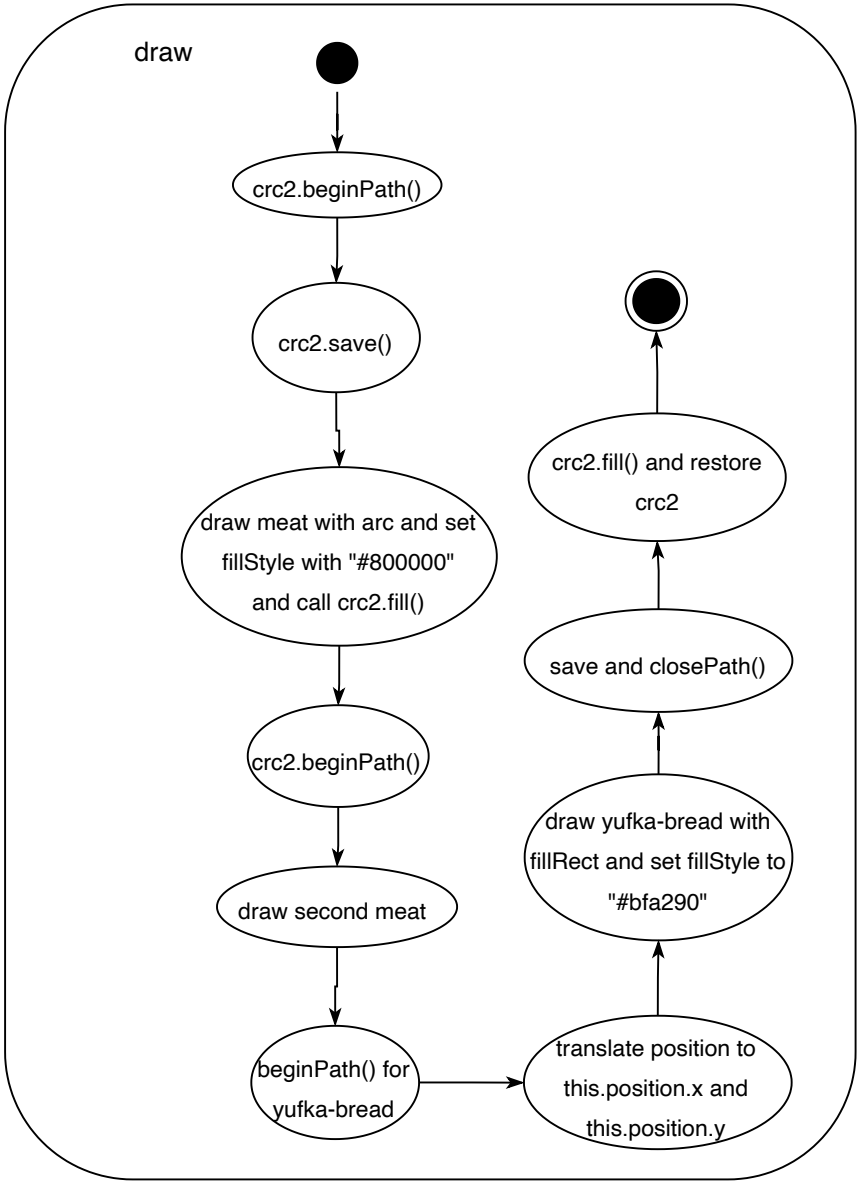
Tomato: Activity Diagram



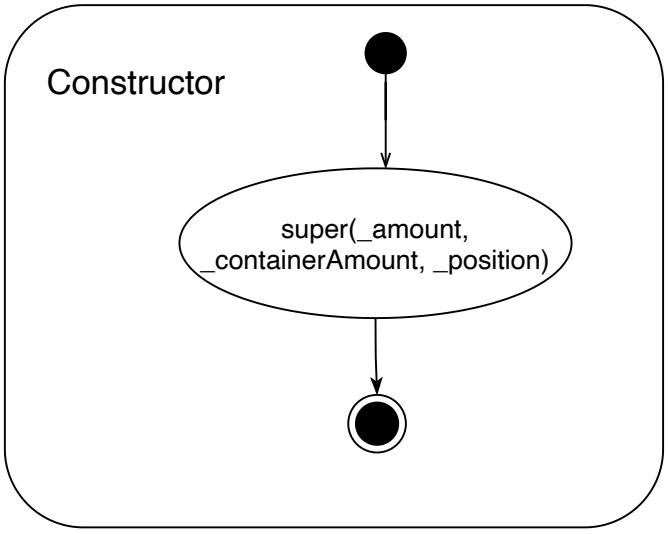


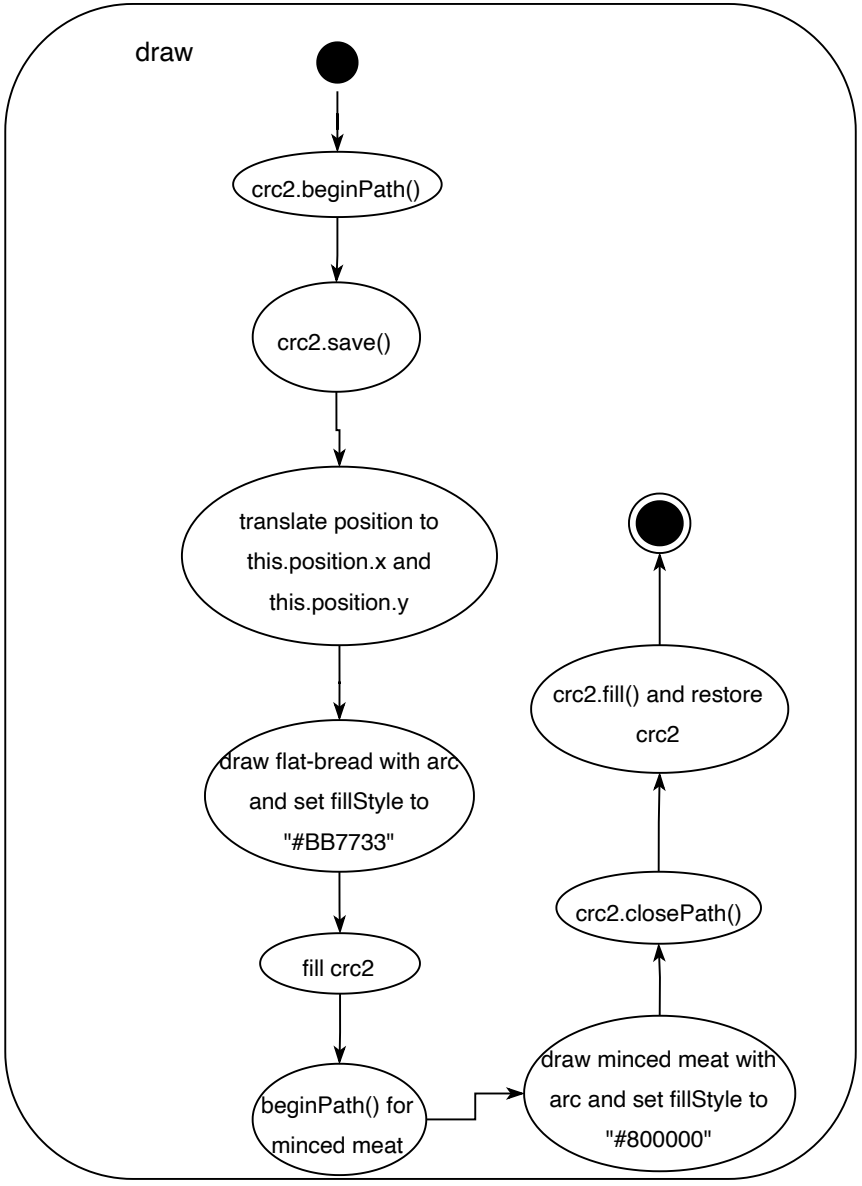
Yufka: Activity Diagram



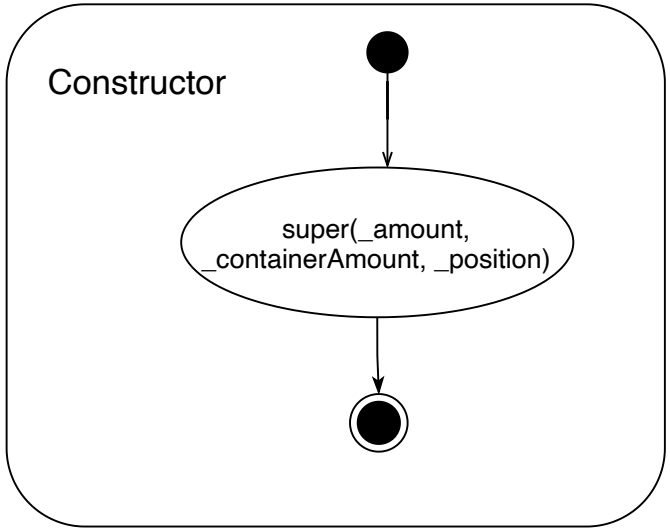


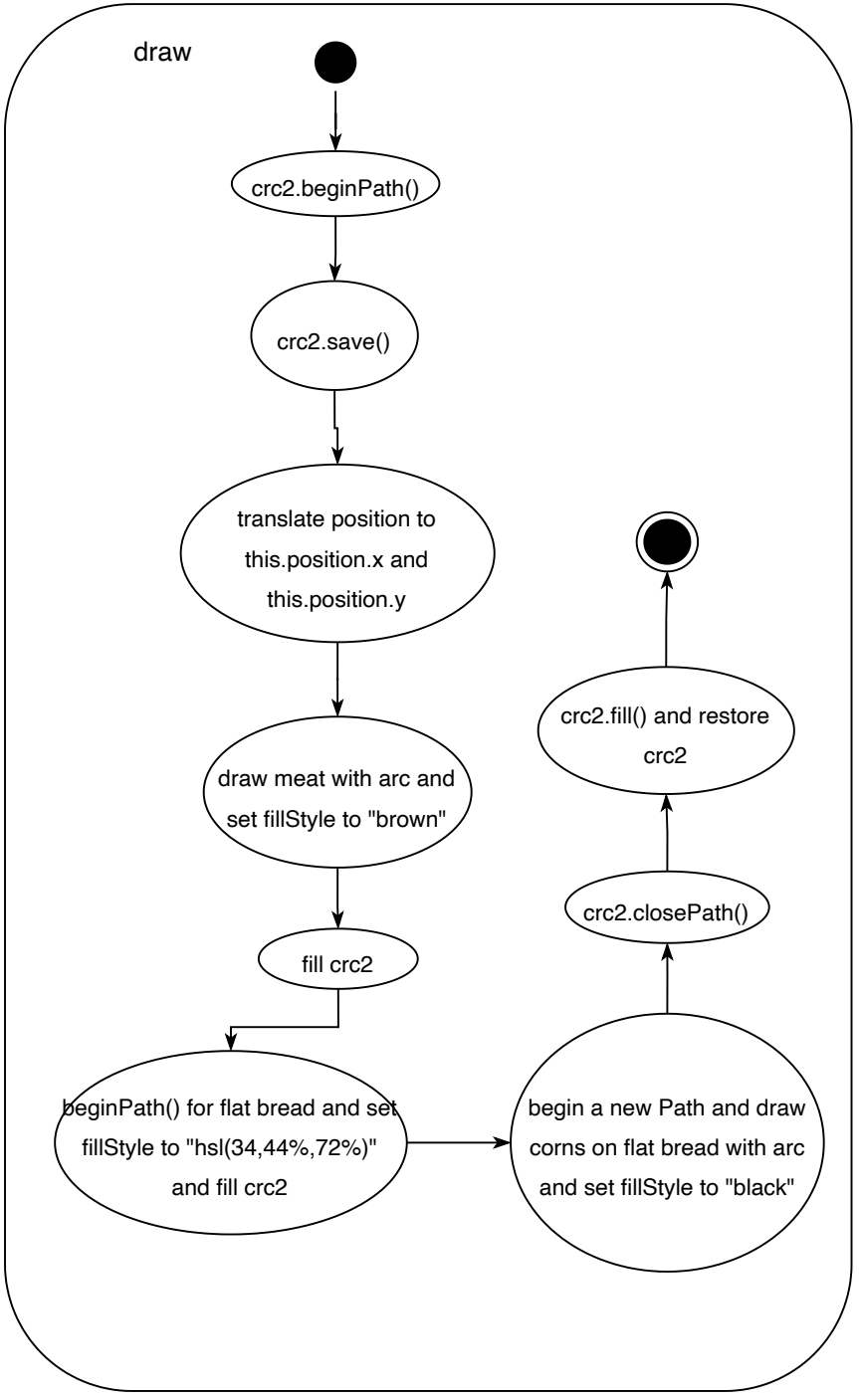
Lahmacun: Activity Diagram



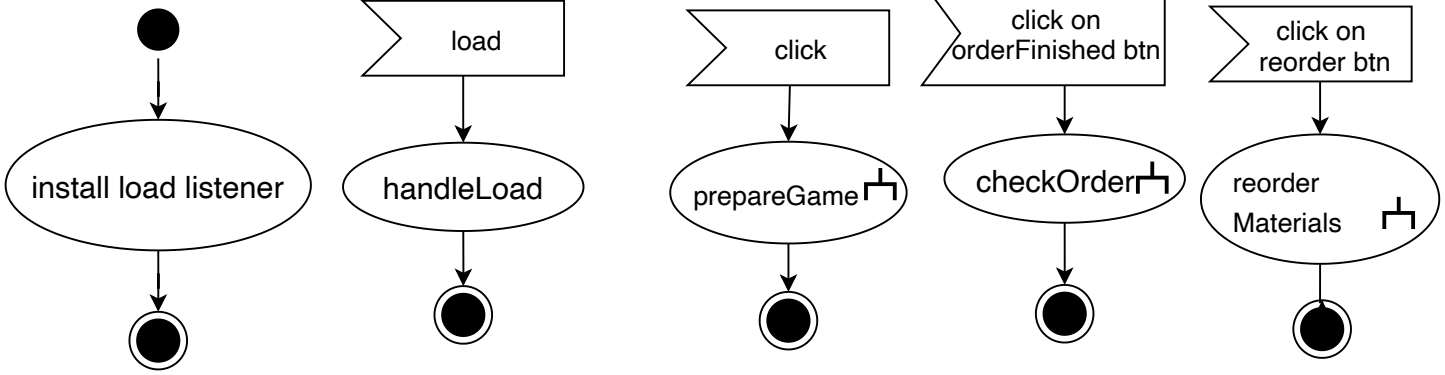


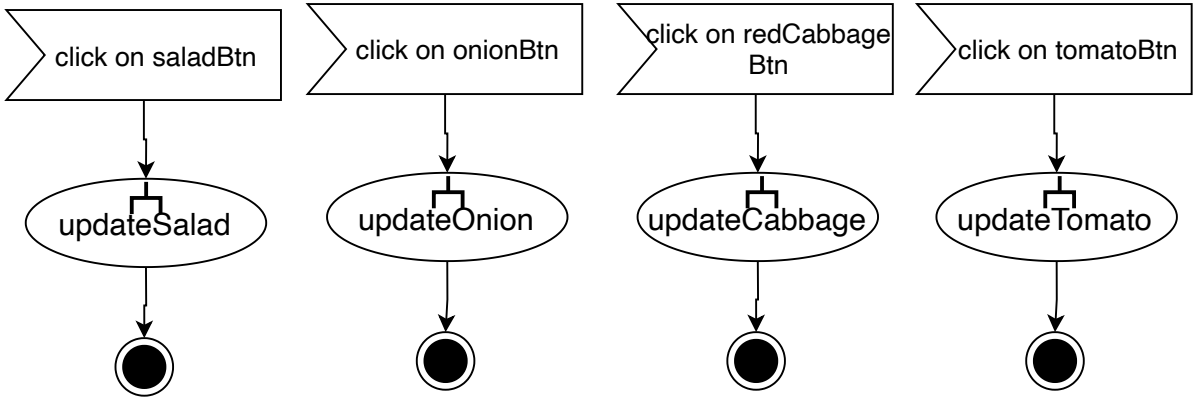
DoenerKebap: Activity Diagram

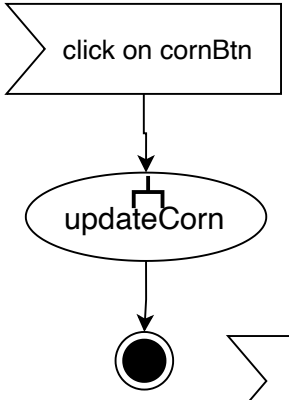


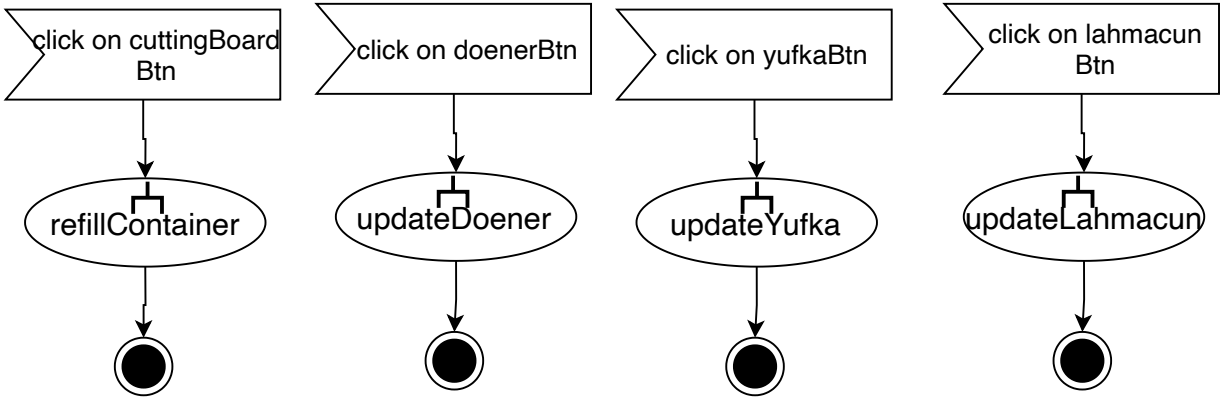


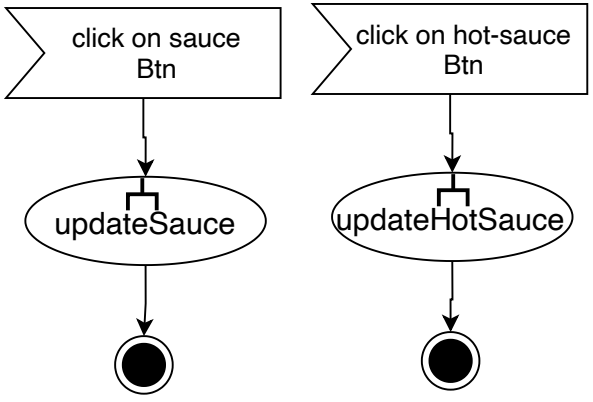
Activity Diagram: Main





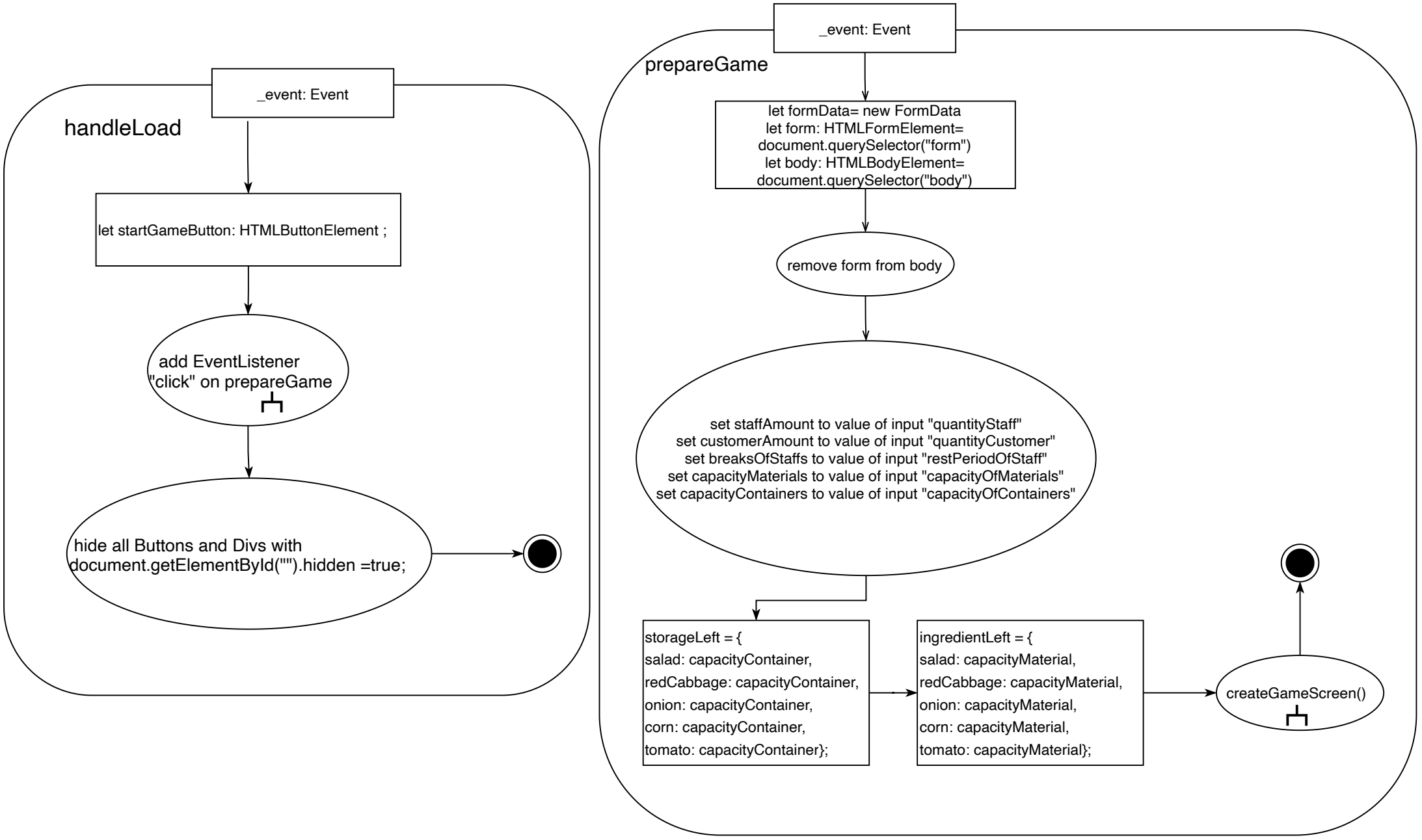


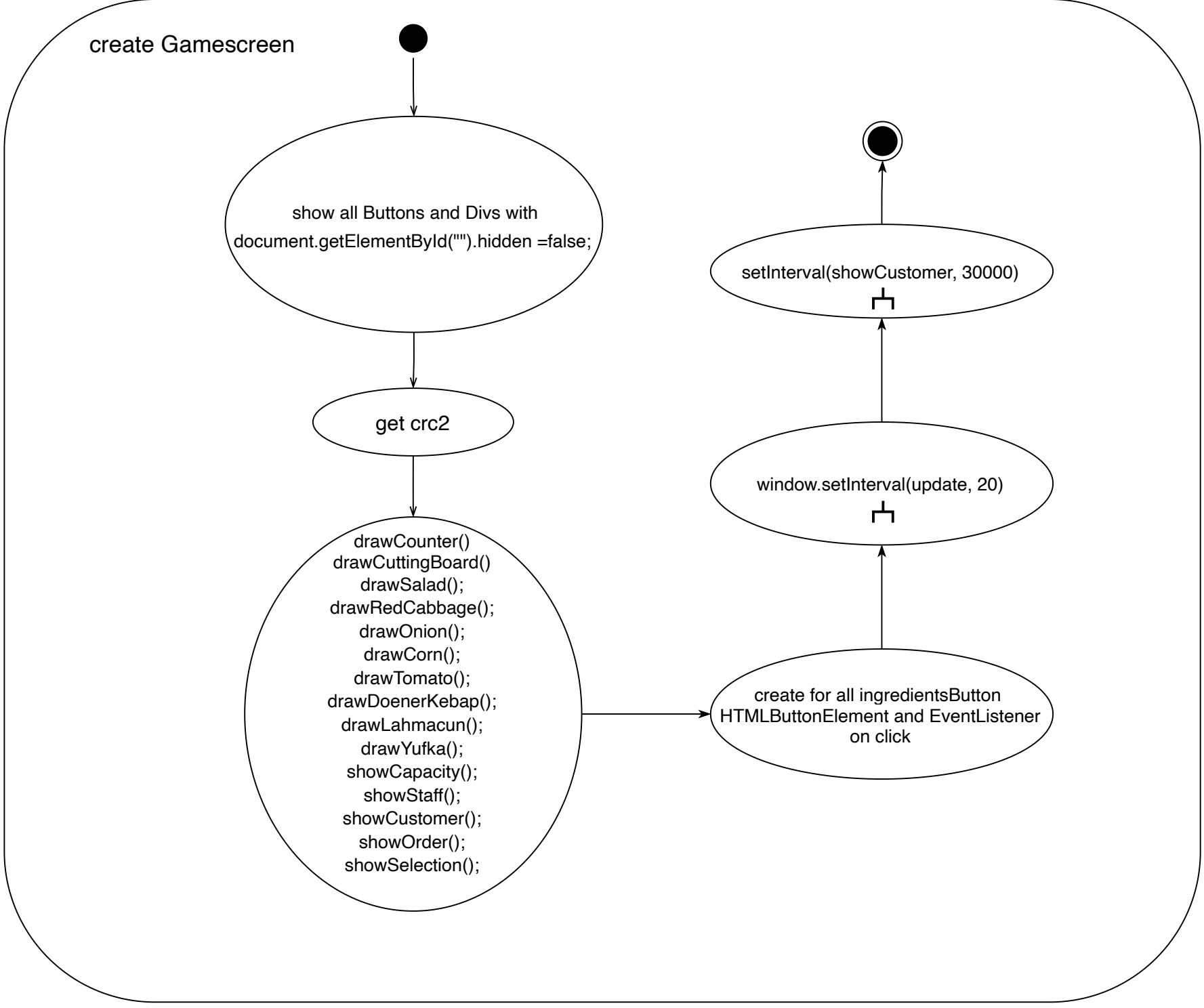


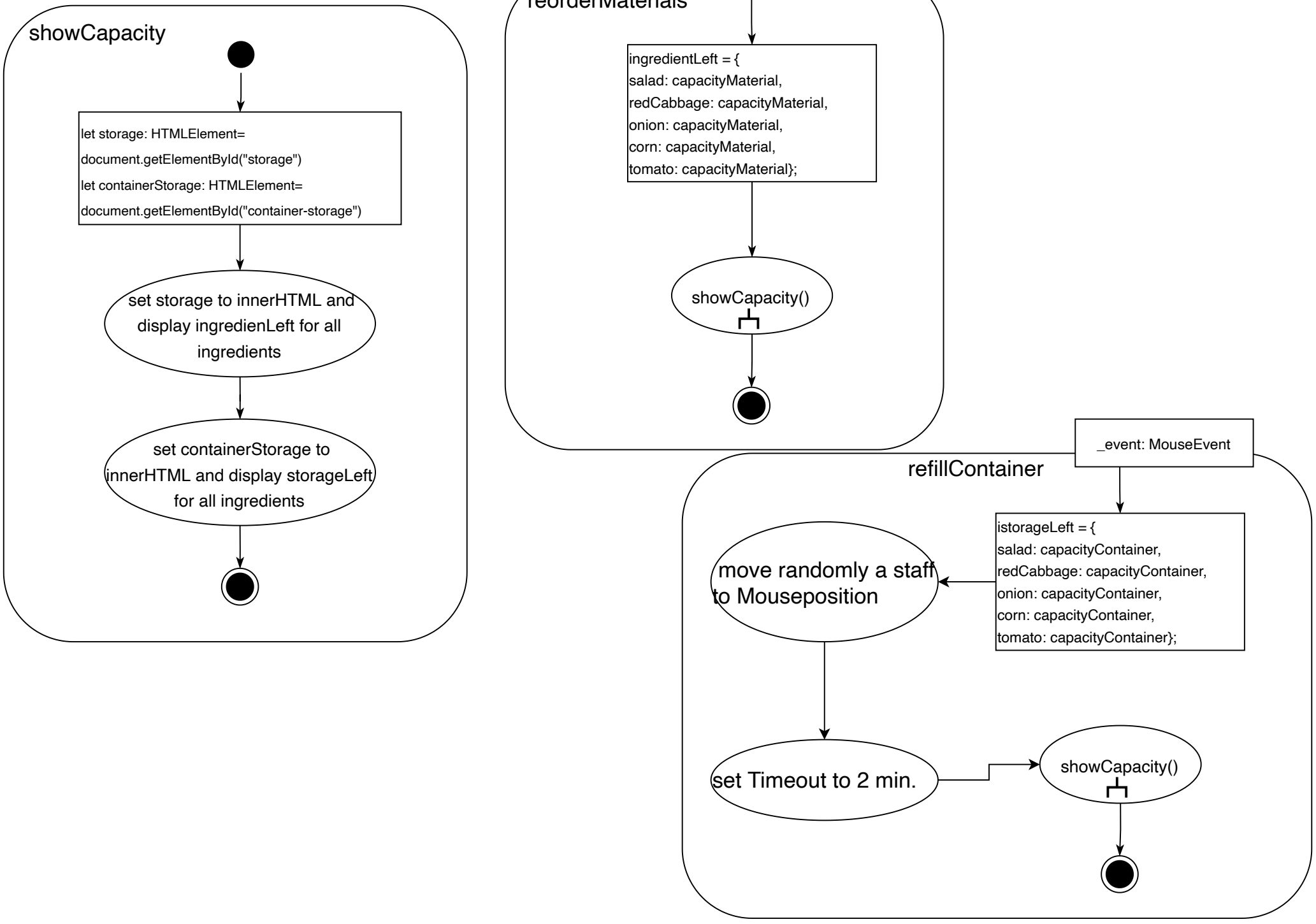


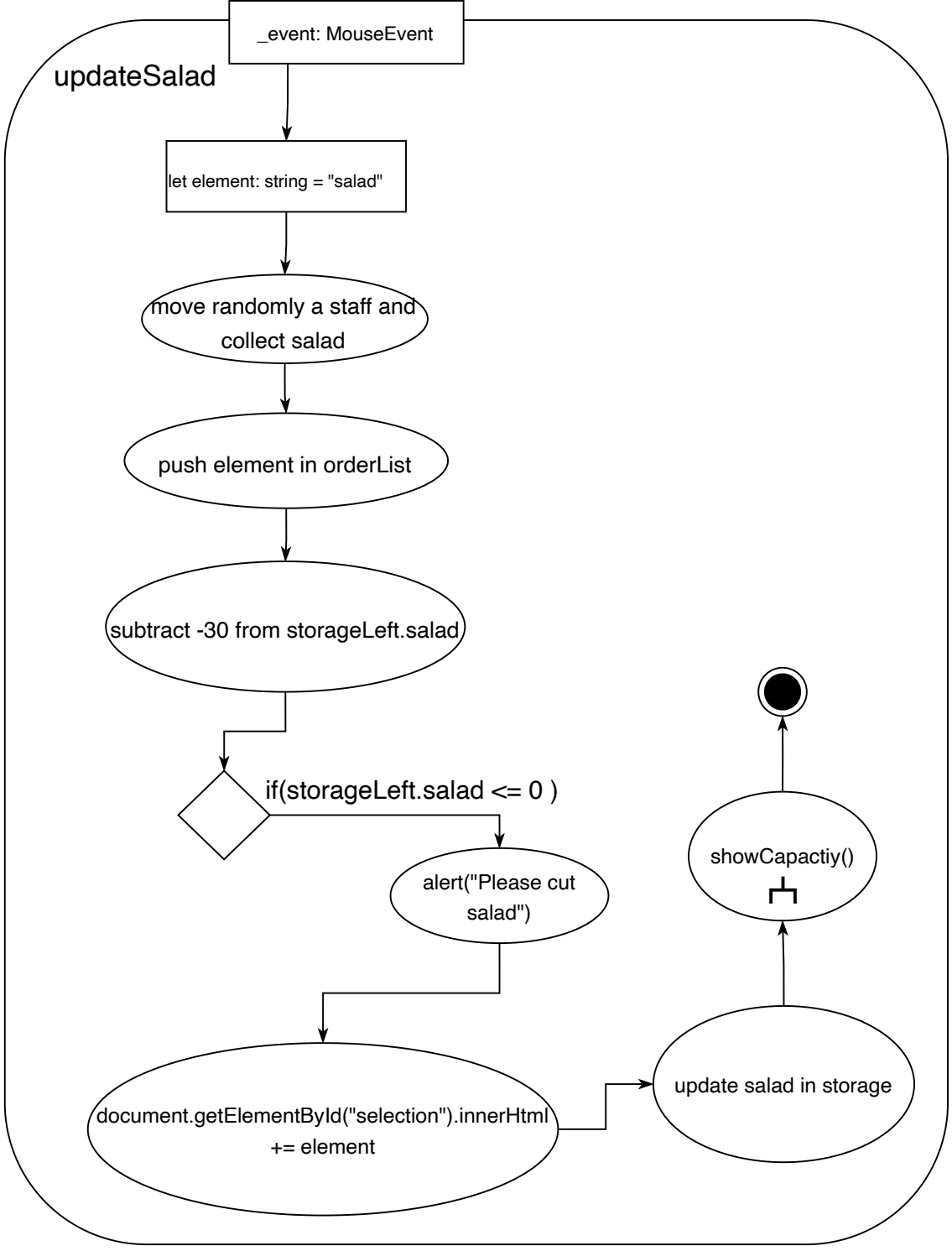

```
let staffAmount: number
let customerAmount: number
let breakofStaff: number
let capacityMaterial: number
let formData: FormData
let orderList: string[] = [];
let request: string[] = [];
let capacityContainer: number;
let staffs: Staff[] = [];
let customers: Customer[] = [];
let ingredients: Ingredient[] = [];
let basis: string[] = ["Döner with meat", "Lahmacun
with minced meat", "Yufka with meat"];
let topping: string[] = ["onion", "salad", "red
cabbage", "corn", "tomato"];
let sauce: string[] = ["sauce", "hot-sauce"];
let storageLeft: Storage;
let ingredientLeft: Storage;
export let crc2: CanvasRenderingContext2D;
export let canvas: HTMLCanvasElement | null;
let staffIsMad: boolean == false;
let customerIsMad: boolean == false;
```

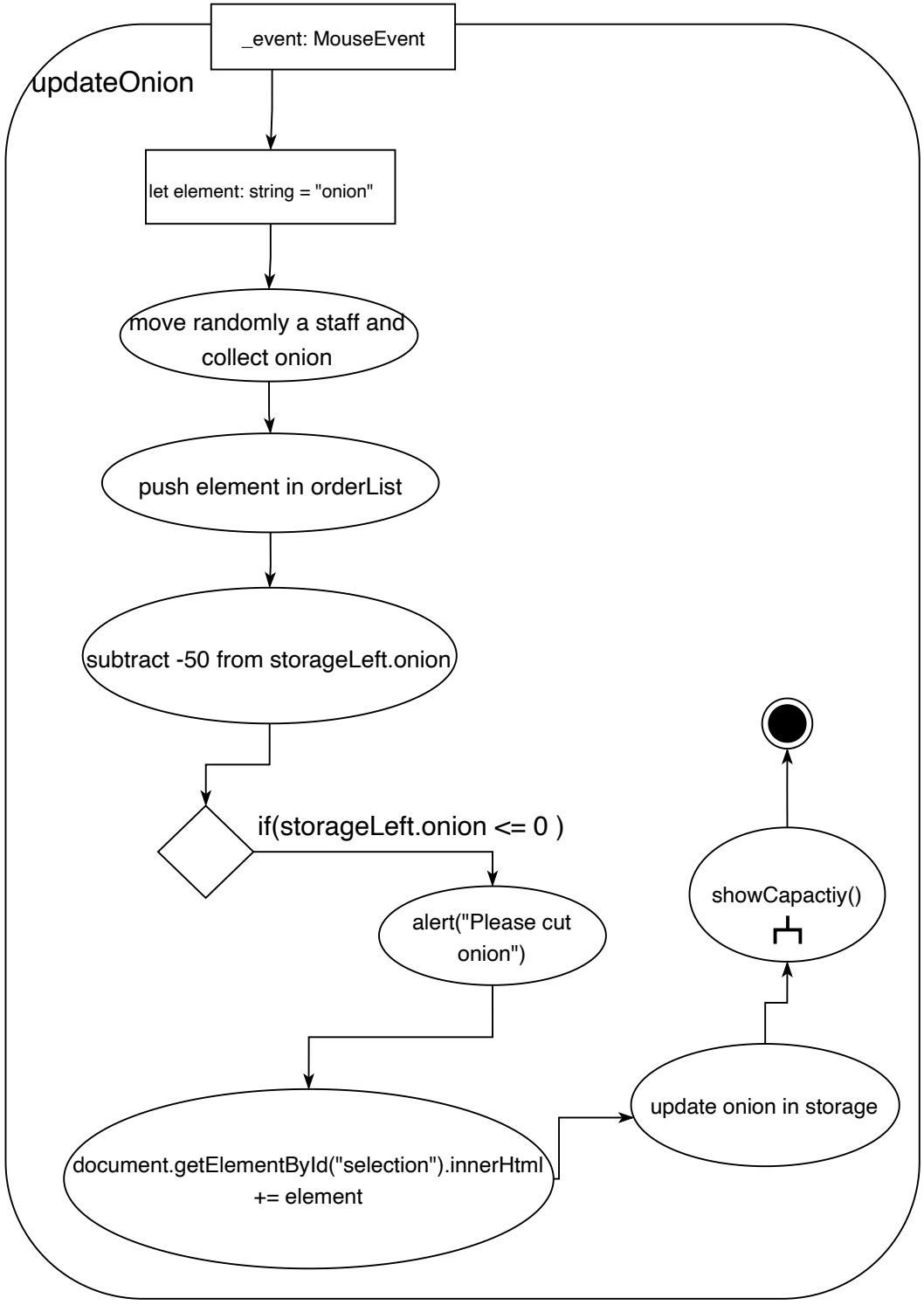
```
interface Storage {
salad: number;
redCabbage: number;
onion: number;
corn: number;
tomato: number;
}
```

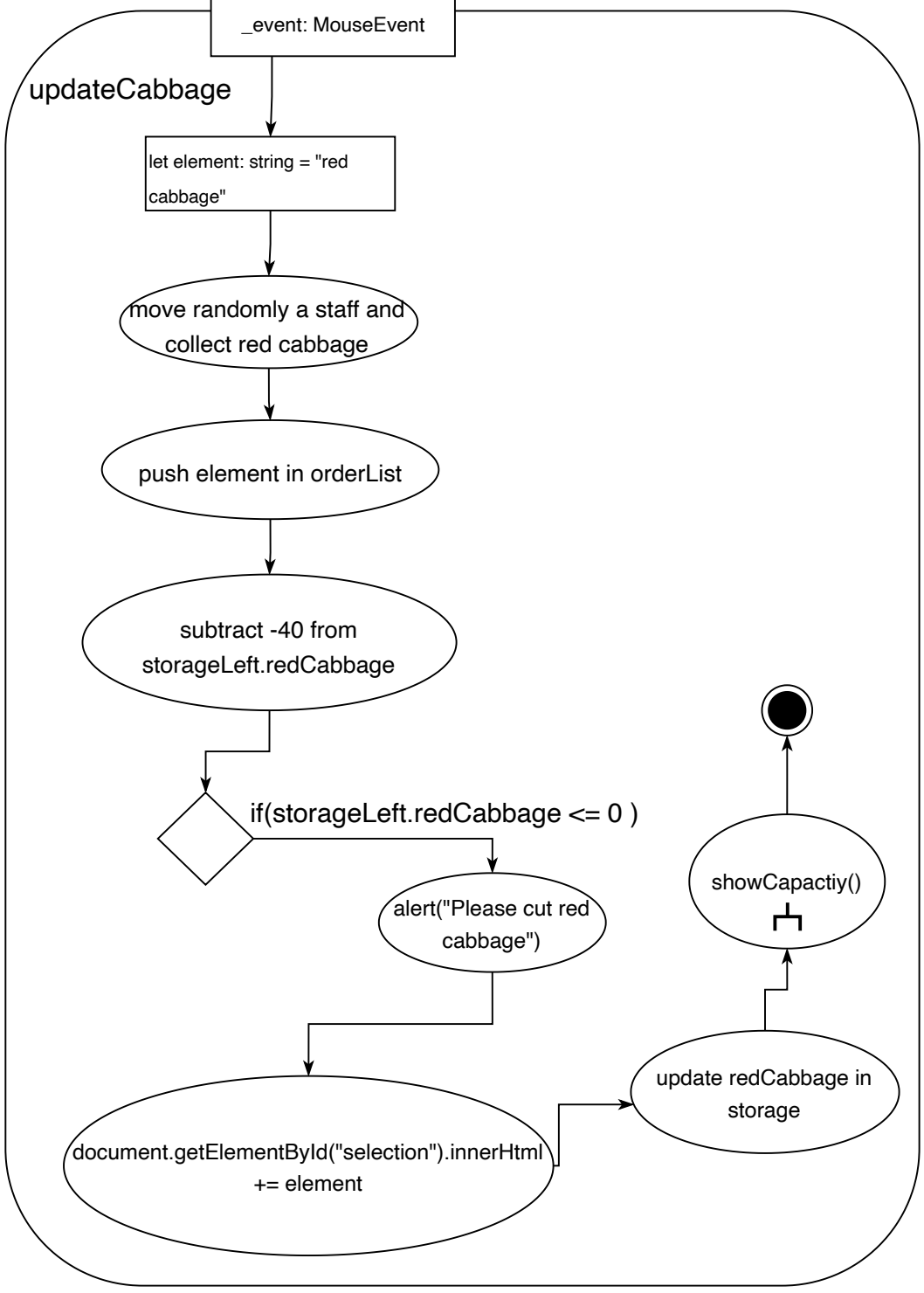

handleLoad
_event: Event
let startGameButton: HTMLButtonElement ;
add EventListener
"click" on prepareGame
hide all Buttons and Divs with
document.getElementById("").hidden =true;
prepareGame
_event: Event
let formData= new FormData
let form: HTMLFormElement=
document.querySelector("form")
let body: HTMLBodyElement=
document.querySelector("body")
remove form from body
set staffAmount to value of input "quantityStaff"
set customerAmount to value of input "quantityCustomer"
set breaksOfStaffs to value of input "restPeriodOfStaff"
set capacityMaterials to value of input "capacityOfMaterials"
set capacityContainers to value of input "capacityOfContainers"
storageLeft = {
salad: capacityContainer,
redCabbage: capacityContainer,
onion: capacityContainer,
corn: capacityContainer,
tomato: capacityContainer};
ingredientLeft = {
salad: capacityMaterial,
redCabbage: capacityMaterial,
onion: capacityMaterial,
corn: capacityMaterial,
tomato: capacityMaterial};
createGameScreen()

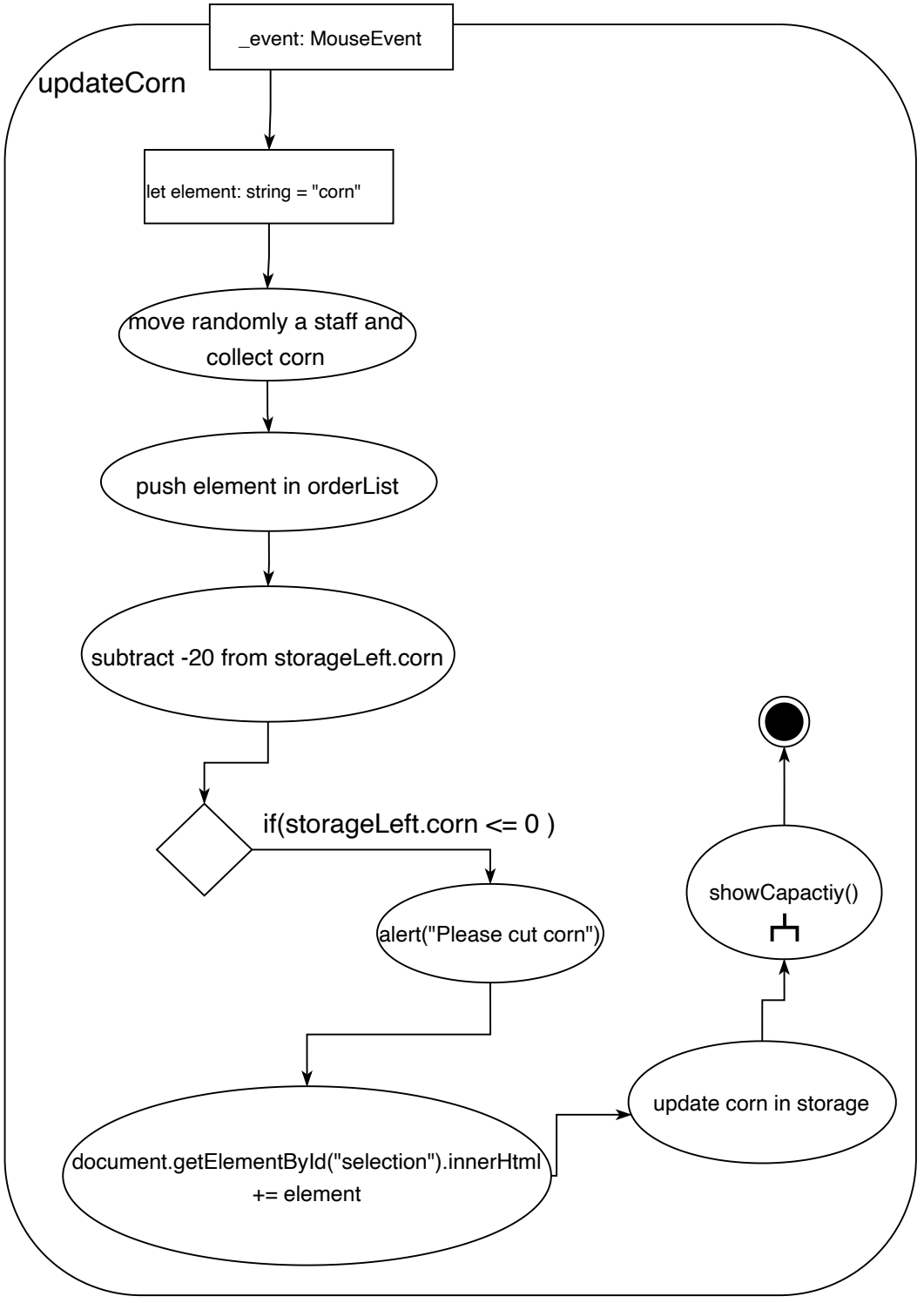

create Gamescreen
show all Buttons and Divs with
document.getElementById("").hidden =false;
get crc2
drawCounter()
drawCuttingBoard()
drawSalad();
drawRedCabbage();
drawOnion();
drawCorn();
drawTomato();
drawDoenerKebap();
drawLahmacun();
drawYufka();
showCapacity();
showStaff();
showCustomer();
showOrder();
showSelection();
create for all ingredientsButton
HTMLButtonElement and EventListener
on click
window.setInterval(update, 20)
setInterval(showCustomer, 30000)

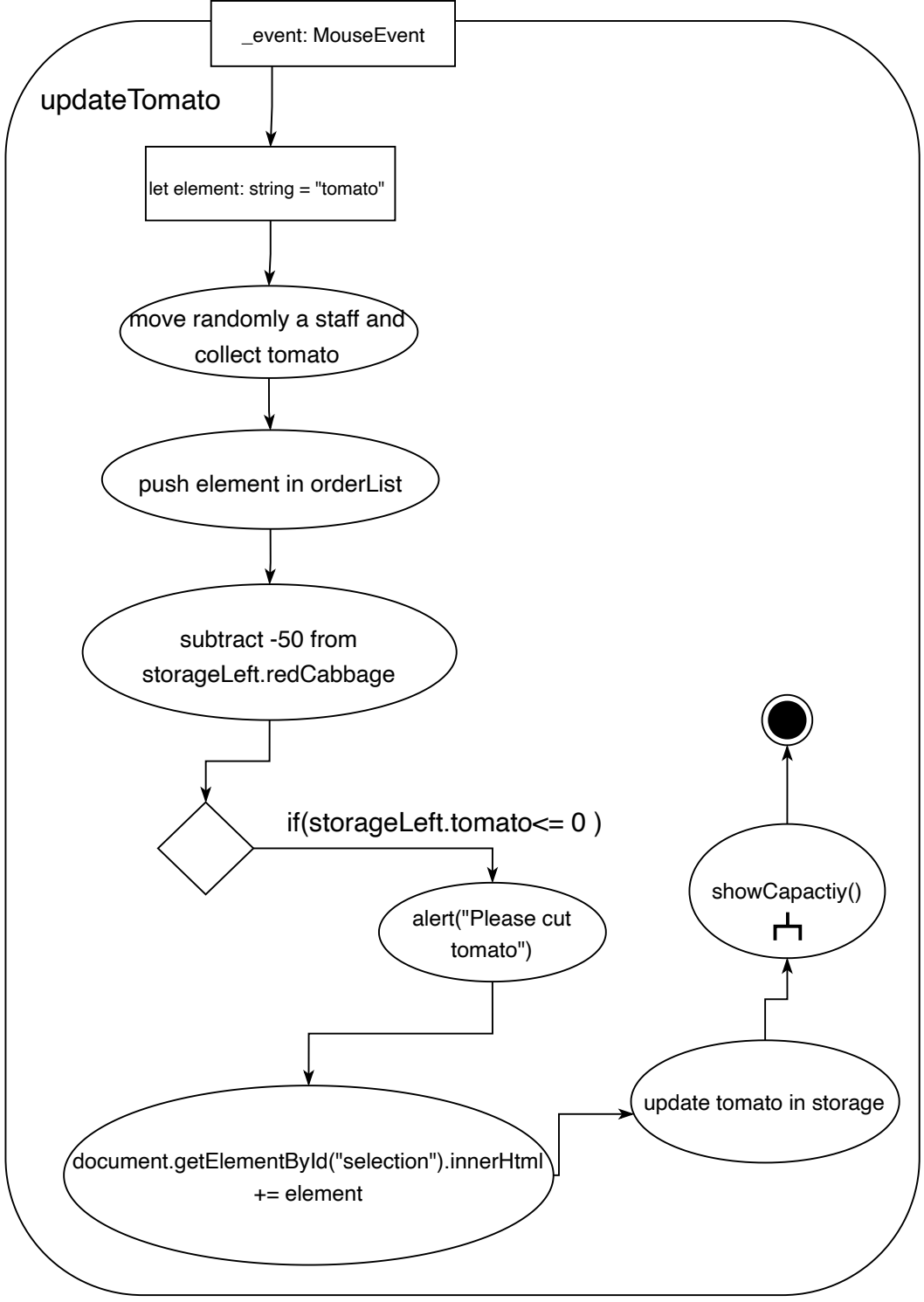

showCapacity
let storage: HTMLElement=
document.getElementById("storage")
let containerStorage: HTMLElement=
document.getElementById("container-storage")
set storage to innerHTML and display ingredienLeft for all ingredients
set containerStorage to innerHTML and display storageLeft for all ingredients
reorderMaterials
_event: Event
ingredientLeft = {
salad: capacityMaterial,
redCabbage: capacityMaterial,
onion: capacityMaterial,
corn: capacityMaterial,
tomato: capacityMaterial};
showCapacity()
refillContainer
_event: MouseEvent
istorageLeft = {
salad: capacityContainer,
redCabbage: capacityContainer,
onion: capacityContainer,
corn: capacityContainer,
tomato: capacityContainer};
move randomly a staff to Mouseposition
set Timeout to 2 min.
showCapacity()

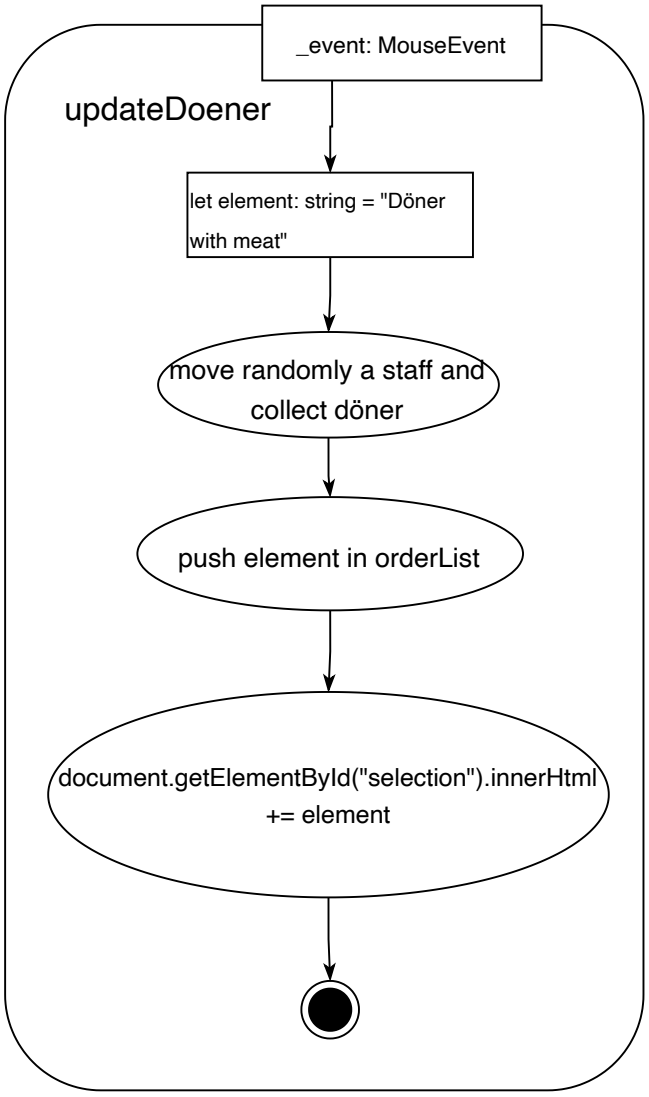

updateSalad
_event: MouseEvent
let element: string = "salad"
move randomly a staff and collect salad
push element in orderList
subtract -30 from storageLeft.salad
if(storageLeft.salad <= 0)
alert("Please cut salad")
showCapactiy()
document.getElementById("selection").innerHtml += element
update salad in storage

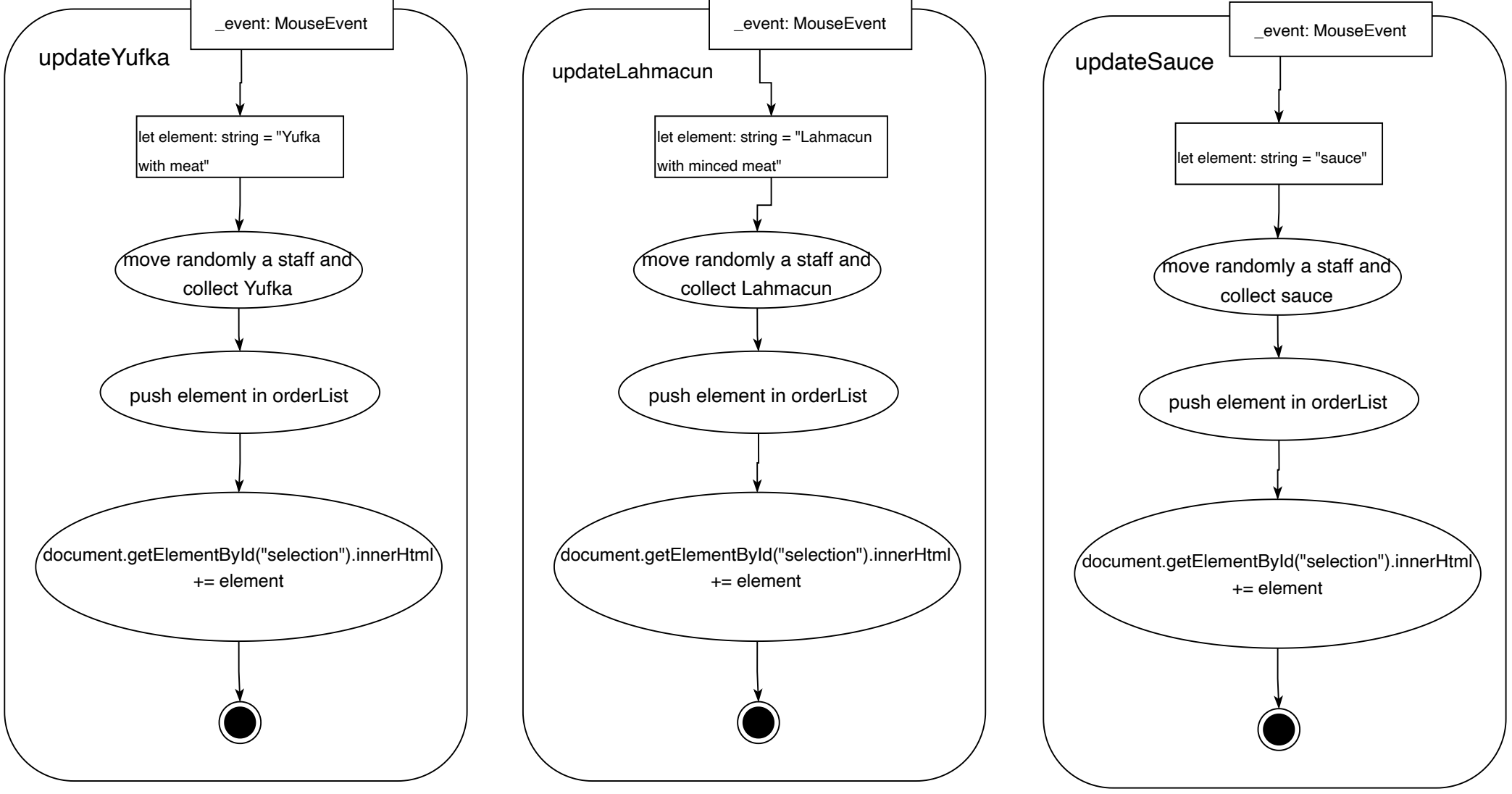

updateOnion
_event: MouseEvent
let element: string = "onion"
move randomly a staff and collect onion
push element in orderList
subtract -50 from storageLeft.onion
if(storageLeft.onion <= 0)
alert("Please cut onion")
showCapactiy()
document.getElementById("selection").innerHtml += element
update onion in storage

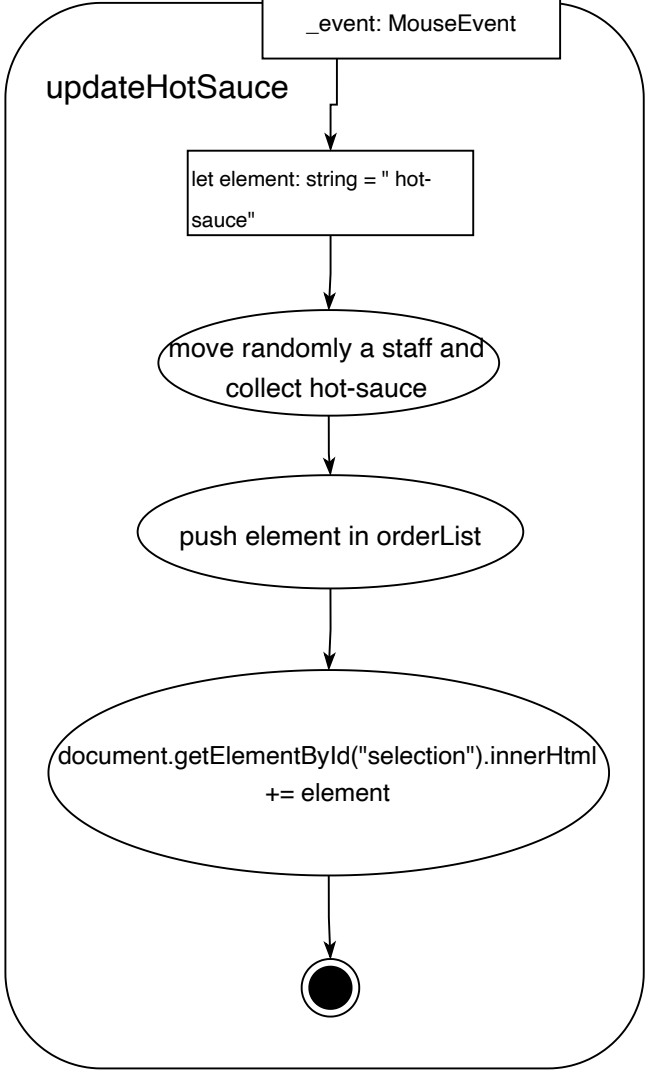

_event: MouseEvent
updateCabbage
let element: string = "red cabbage"
move randomly a staff and collect red cabbage
push element in orderList
subtract -40 from storageLeft.redCabbage
if(storageLeft.redCabbage <= 0)
alert("Please cut red cabbage")
showCapactiy()
update redCabbage in storage
document.getElementById("selection").innerHtml += element

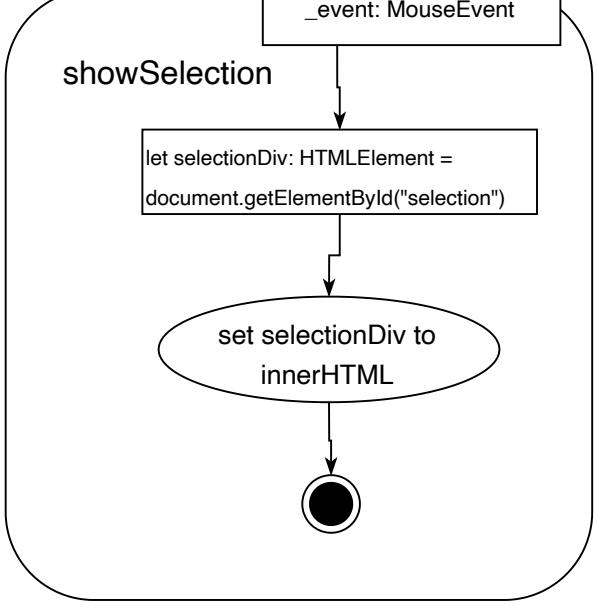

_event: MouseEvent
updateCorn
let element: string = "corn"
move randomly a staff and collect corn
push element in orderList
subtract -20 from storageLeft.corn
if(storageLeft.corn <= 0)
alert("Please cut corn")
showCapactiy()
update corn in storage
document.getElementById("selection").innerHtml += element

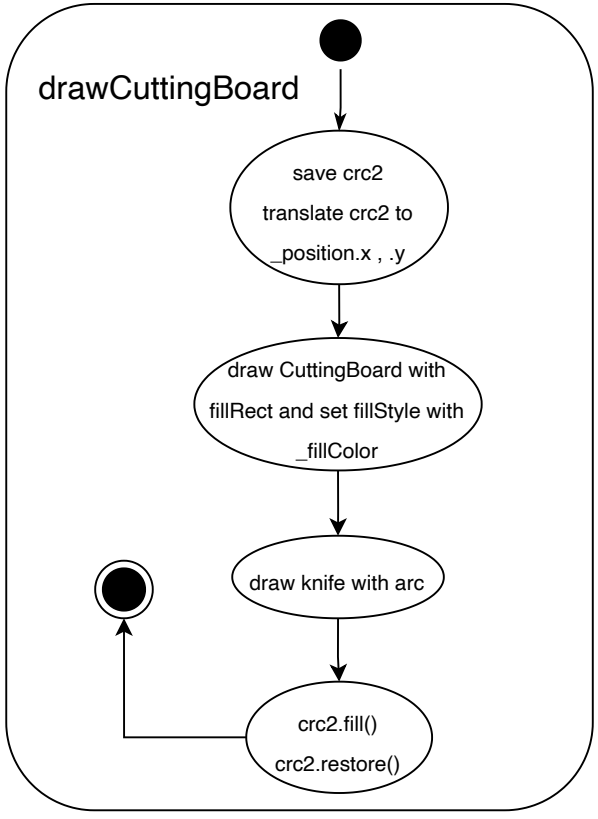

_event: MouseEvent
updateTomato
let element: string = "tomato"
move randomly a staff and collect tomato
push element in orderList
subtract -50 from storageLeft.redCabbage
if(storageLeft.tomato<= 0)
alert("Please cut tomato")
showCapactiy()
update tomato in storage
document.getElementById("selection").innerHtml += element

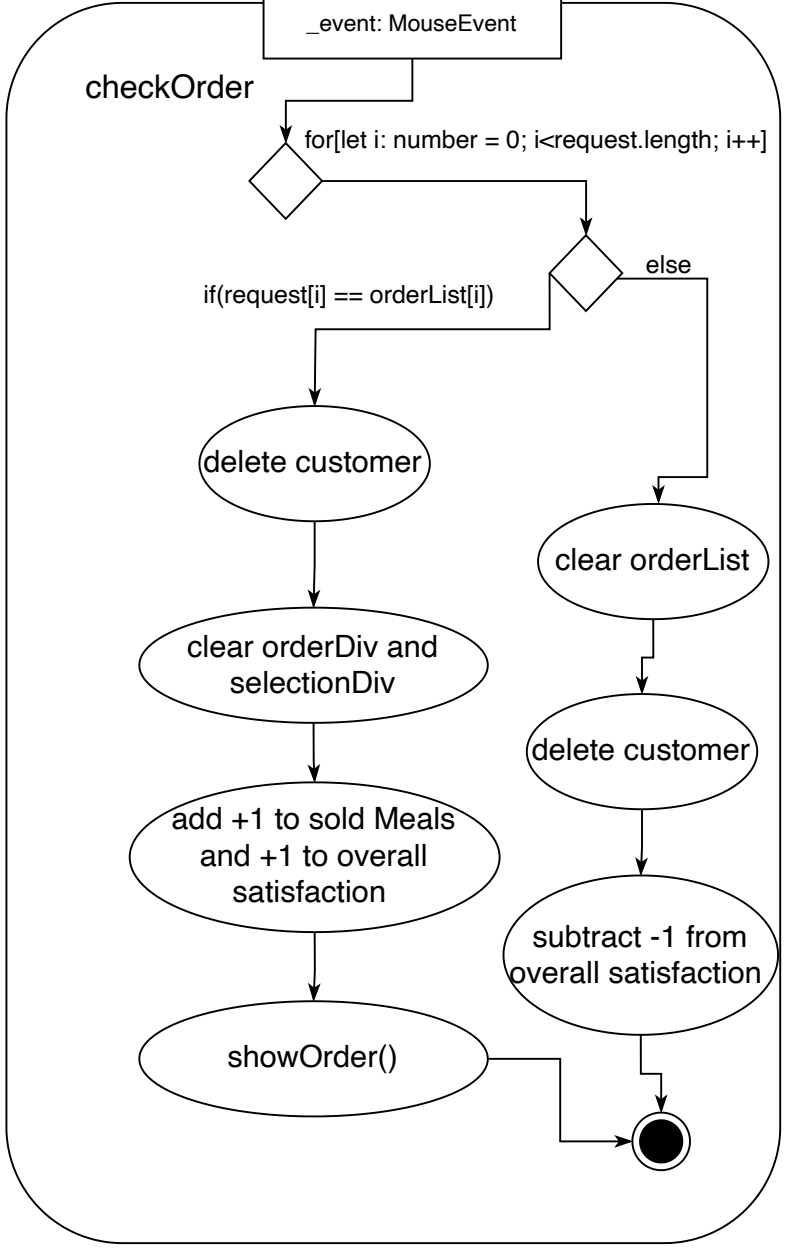

_event: MouseEvent
updateDoener
let element: string = "Döner with meat"
move randomly a staff and collect döner
push element in orderList
document.getElementById("selection").innerHtml += element

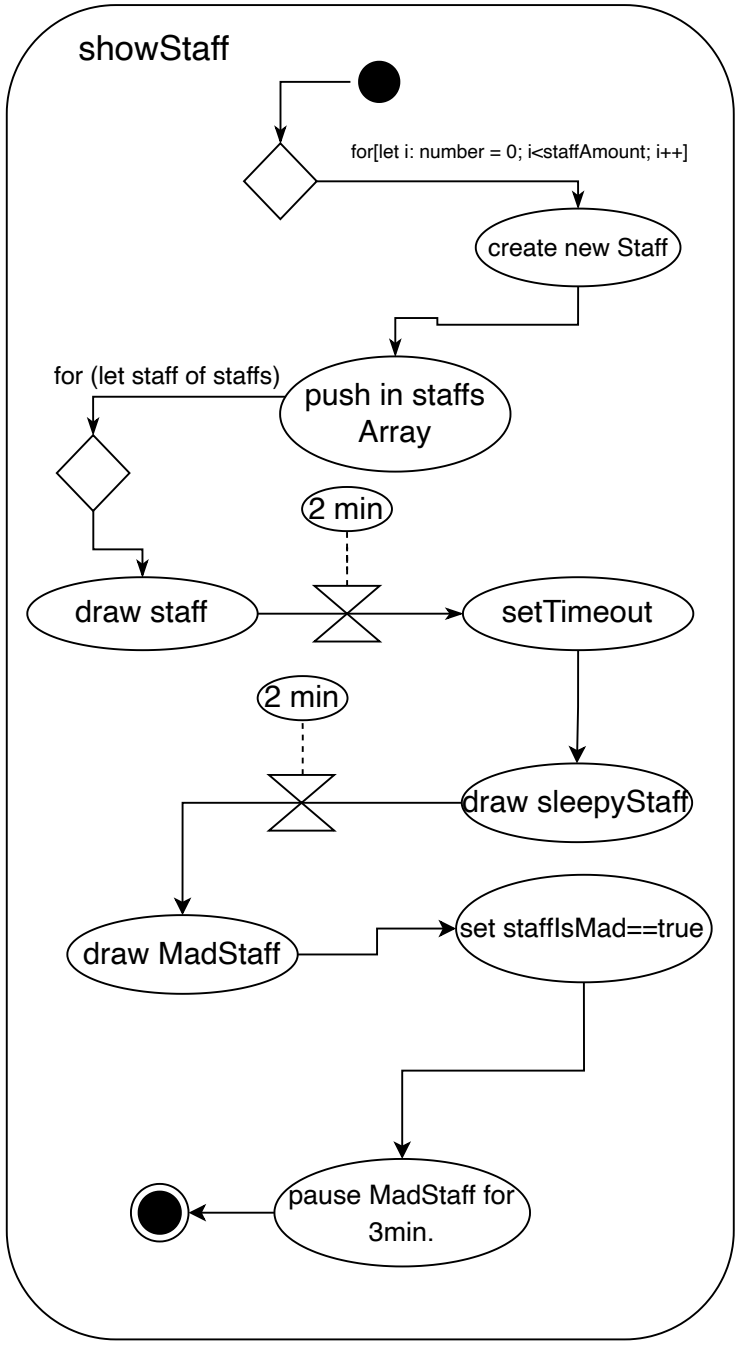

updateYufka
_event: MouseEvent
let element: string = "Yufka with meat"
move randomly a staff and collect Yufka
push element in orderList
document.getElementById("selection").innerHtml += element
updateLahmacun
_event: MouseEvent
let element: string = "Lahmacun with minced meat"
move randomly a staff and collect Lahmacun
push element in orderList
document.getElementById("selection").innerHtml += element
updateSauce
_event: MouseEvent
let element: string = "sauce"
move randomly a staff and collect sauce
push element in orderList
document.getElementById("selection").innerHtml += element

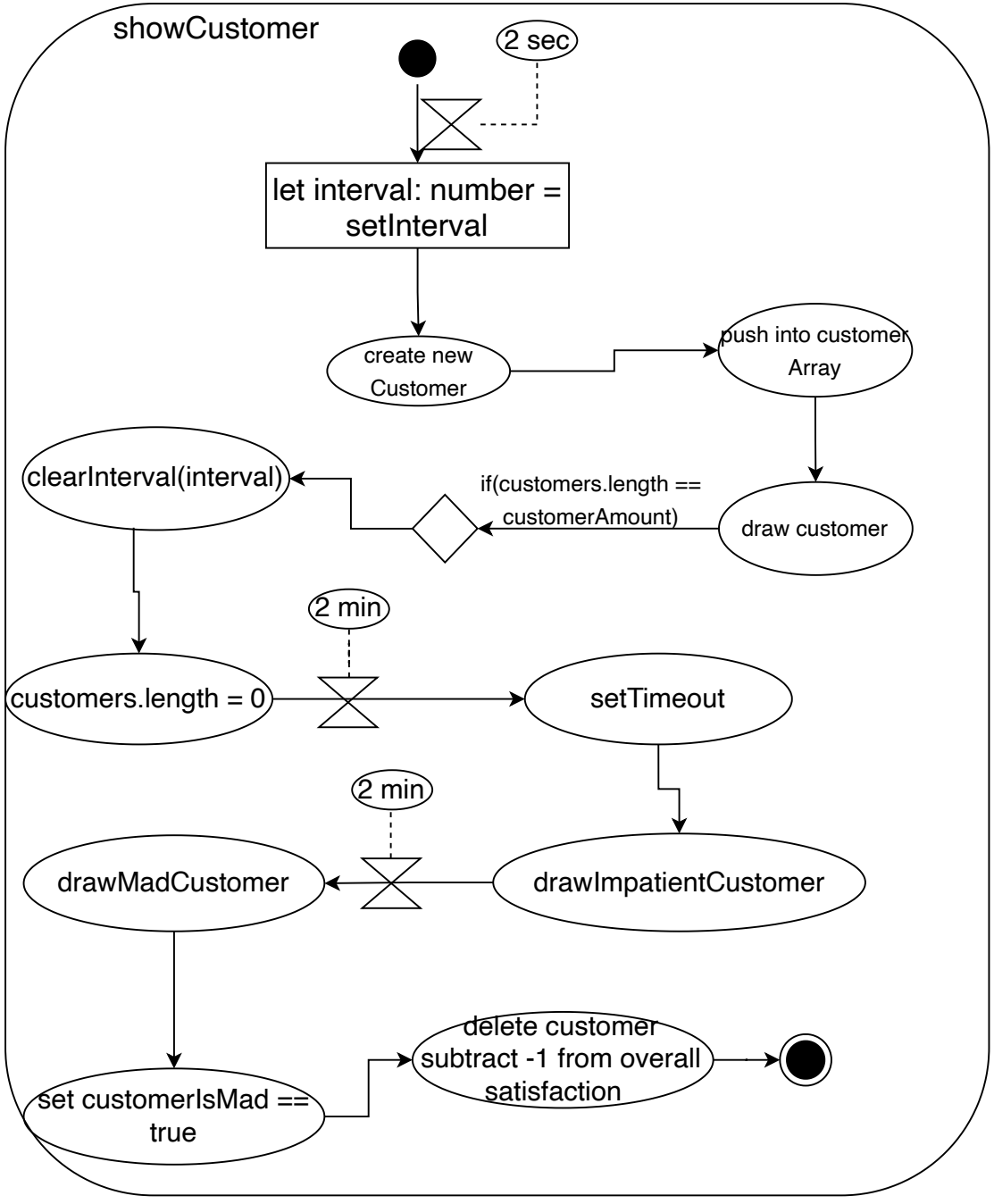

_event: MouseEvent
updateHotSauce
let element: string = " hot-sauce"
move randomly a staff and collect hot-sauce
push element in orderList
document.getElementById("selection").innerHtml += element

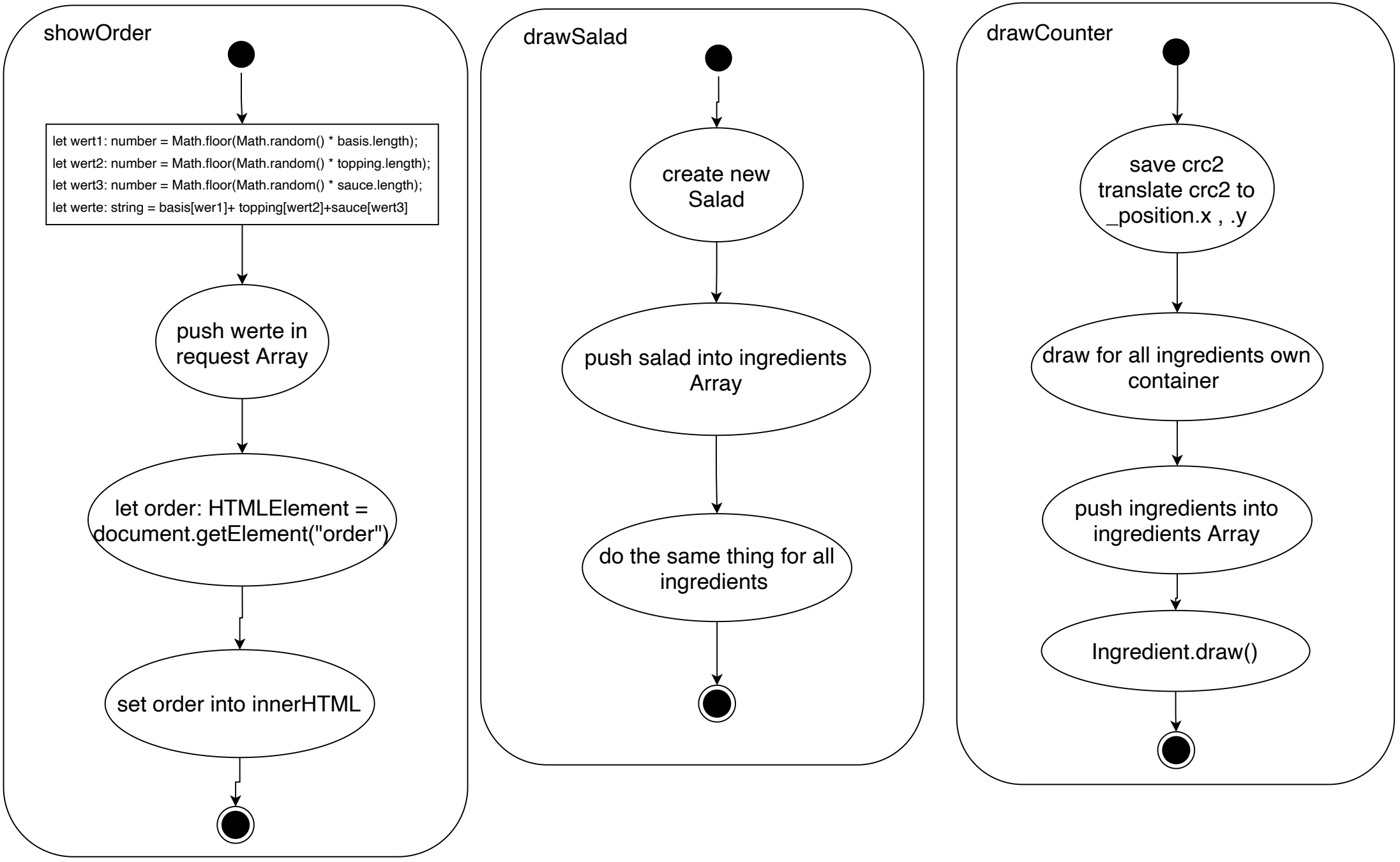

_event: MouseEvent
showSelection
let selectionDiv: HTMLElement = document.getElementById("selection")
set selectionDiv to innerHTML


drawCuttingBoard
save crc2
translate crc2 to _position.x , .y
draw CuttingBoard with fillRect and set fillStyle with _fillColor
draw knife with arc
crc2.fill()
crc2.restore()


_event: MouseEvent
checkOrder
for[let i: number = 0; i<request.length; i++]
if(request[i] == orderList[i])
else
delete customer
clear orderList
clear orderDiv and selectionDiv
delete customer
add +1 to sold Meals and +1 to overall satisfaction
subtract -1 from overall satisfaction
showOrder()


showStaff
for[let i: number = 0; i<staffAmount; i++]
create new Staff
for (let staff of staffs)
push in staffs Array
2 min
draw staff
setTimeout
2 min
draw sleepyStaff
draw MadStaff
set staffIsMad==true
pause MadStaff for 3min.


showCustomer
2 sec
let interval: number = setInterval
create new Customer
push into customer Array
clearInterval(interval)
if(customers.length == customerAmount)
draw customer
2 min
customers.length = 0
setTimeout
2 min
drawMadCustomer
drawImpatientCustomer
set customerIsMad == true
delete customer subtract -1 from overall satisfaction


showOrder
let wert1: number = Math.floor(Math.random() * basis.length);
let wert2: number = Math.floor(Math.random() * topping.length);
let wert3: number = Math.floor(Math.random() * sauce.length);
let werte: string = basis[wer1]+ topping[wert2]+sauce[wert3]
push werte in request Array
let order: HTMLElement = document.getElement("order")
set order into innerHTML
drawSalad
create new Salad
push salad into ingredients Array
do the same thing for all ingredients
drawCounter
save crc2 translate crc2 to _position.x , .y
draw for all ingredients own container
push ingredients into ingredients Array
Ingredient.draw()