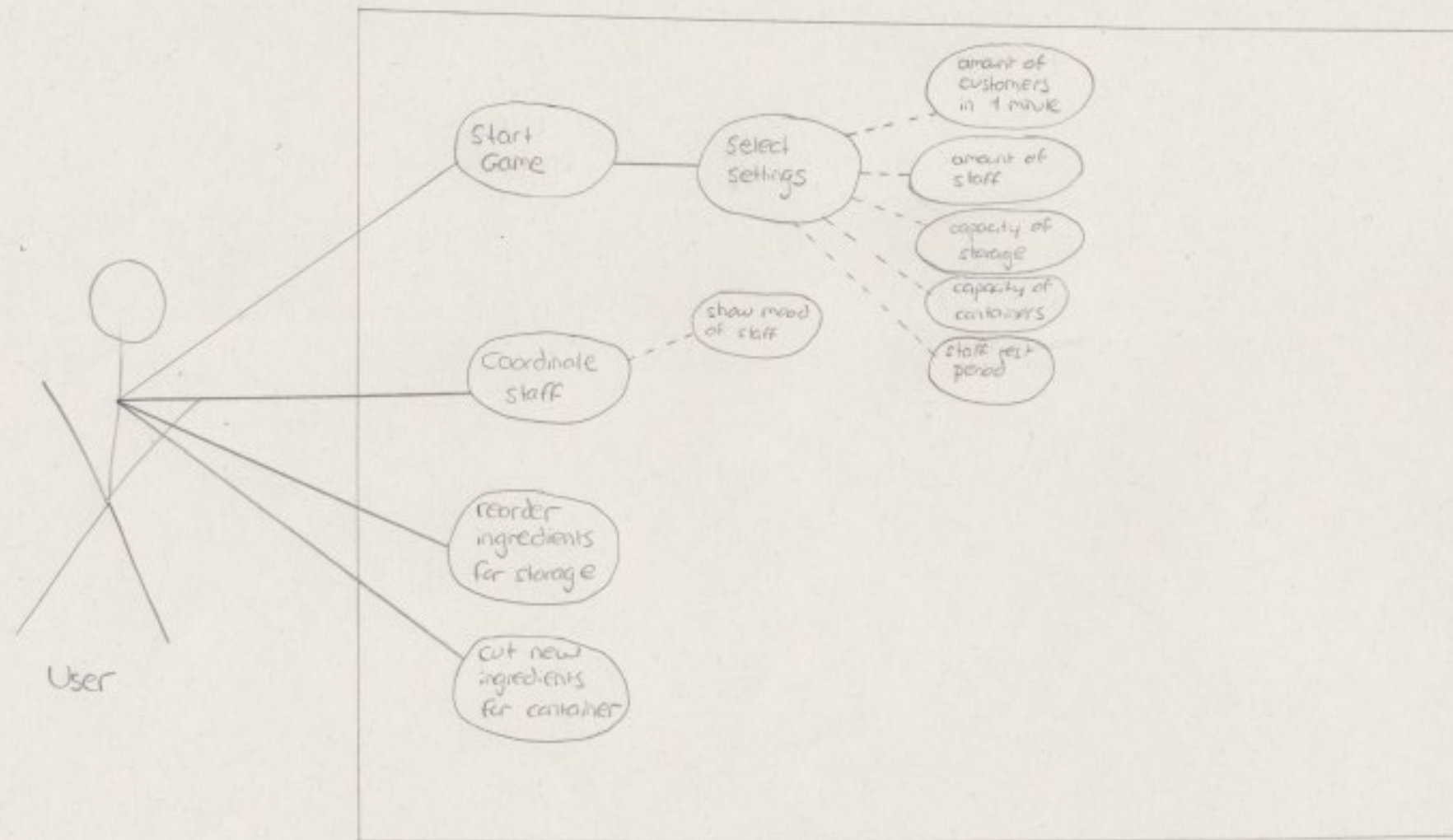


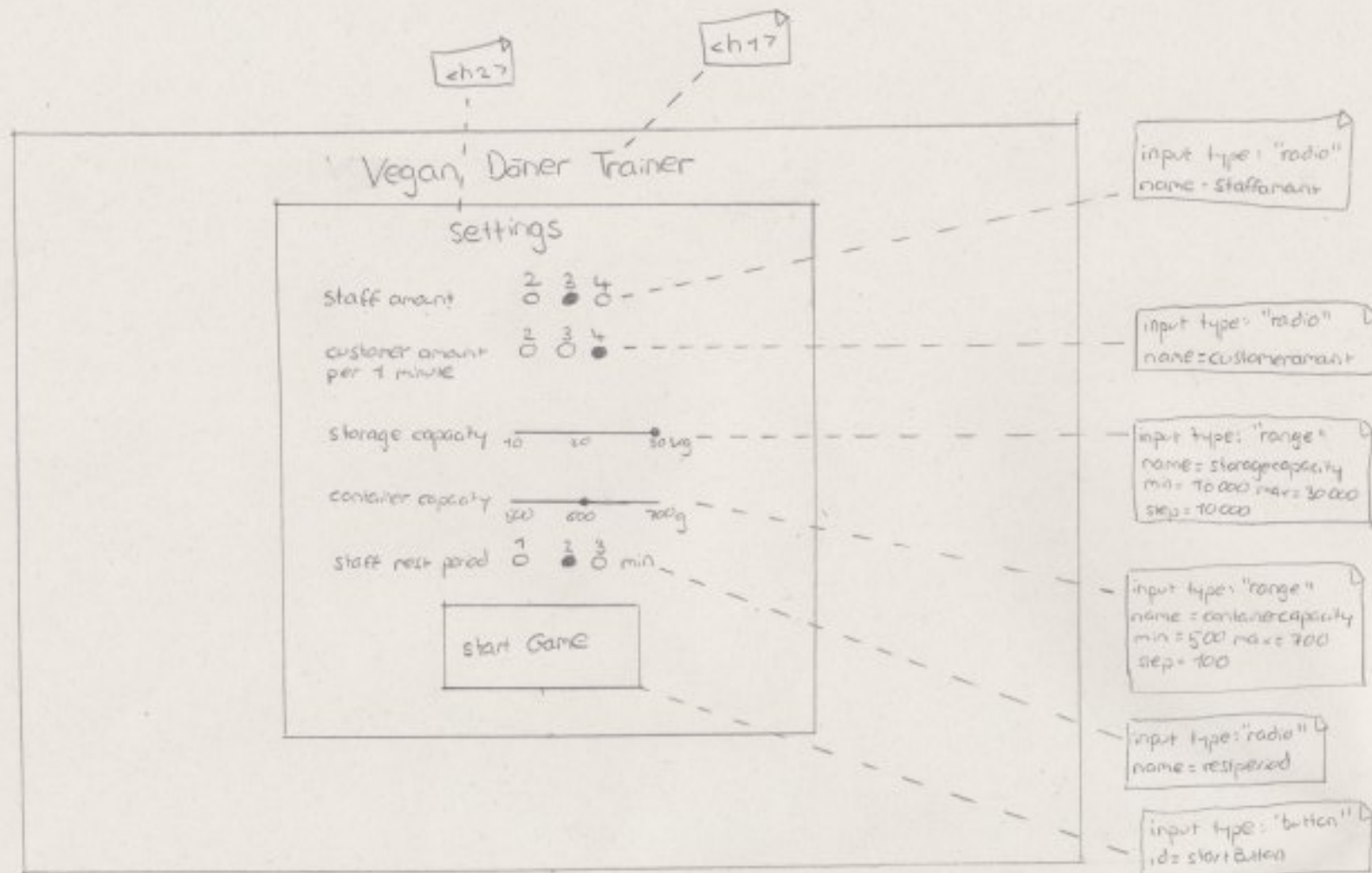
Endabgabe EIA2 – Döner-Trainer

Konzept

Verena Rothweiler
Matrikelnummer: 270156

in Zusammenarbeit mit Neslisah Koc





Vegan Döner Trainer: Scribble

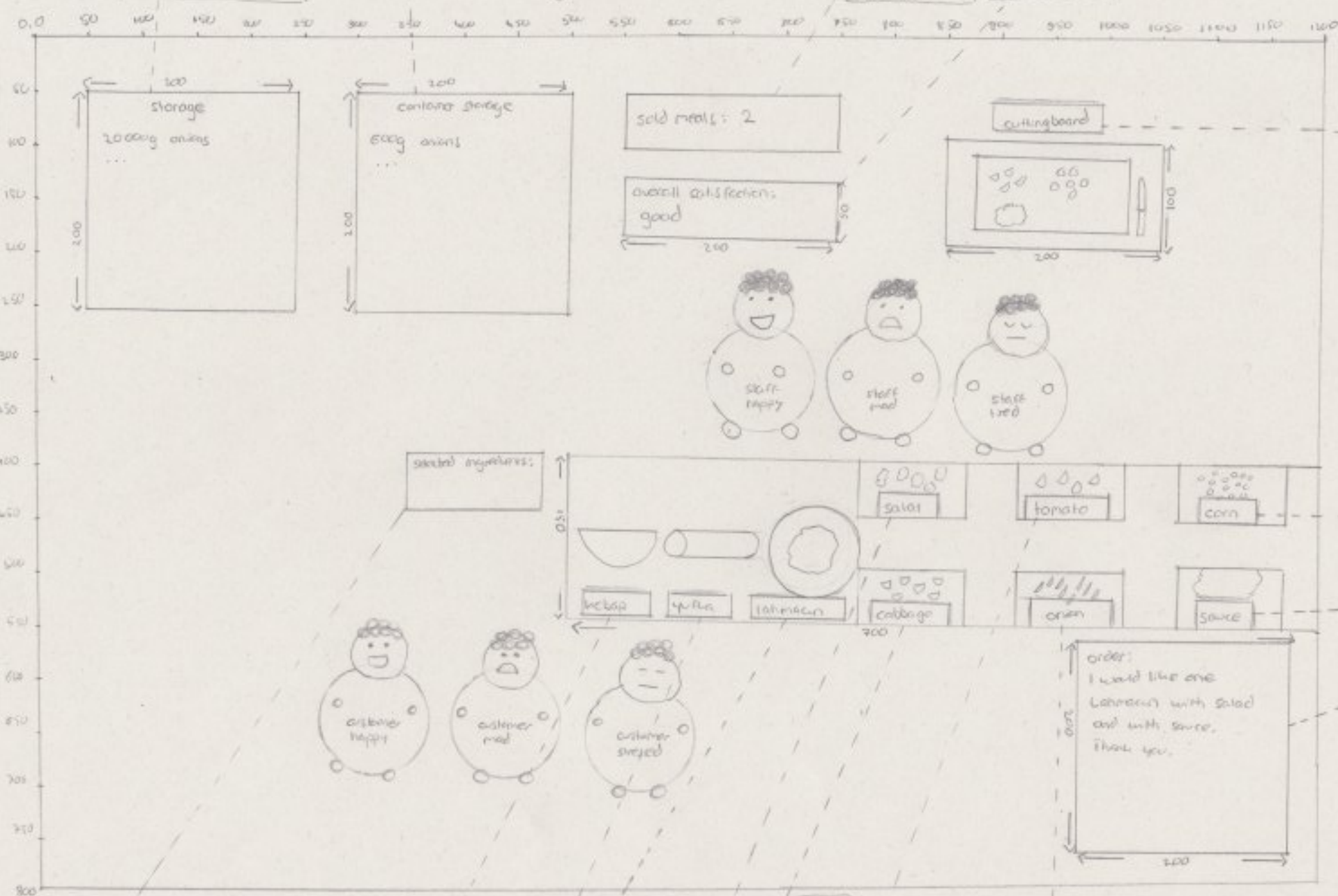
<div id="storage"></div>

<div id="container"></div>

Vegan Döner Trainer

<div id="soldmeals"></div>

<div id="overall"></div>



<button id="cuttingboard"></button>

<button id="corn"></button>

<button id="sauce"></button>

<div id="order"></div>

<div id="selectedingredients"></div>

<button id="beetroot"></button>

<button id="tahin"></button>

<button id="cabbage"></button>

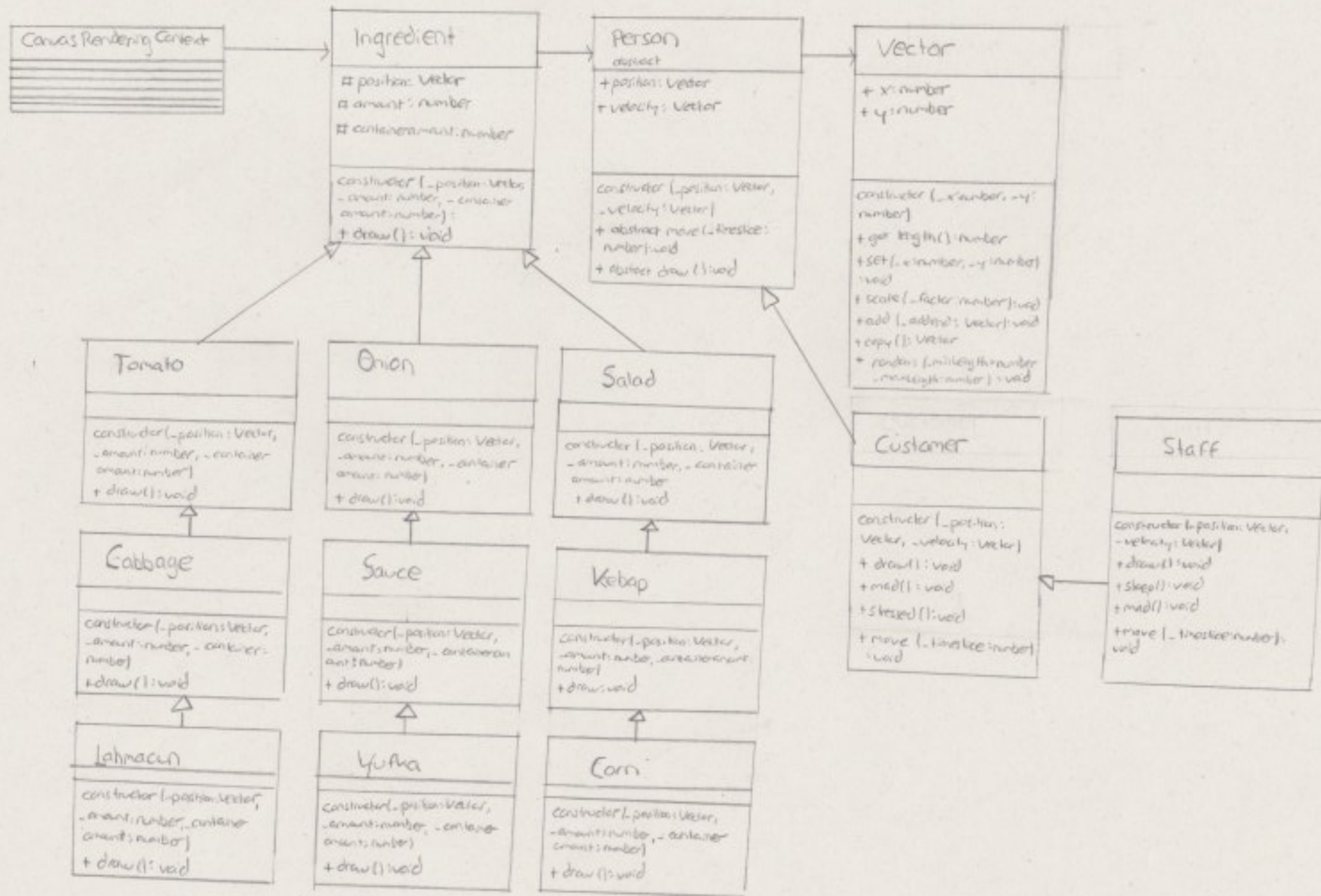
<button id="onion"></button>

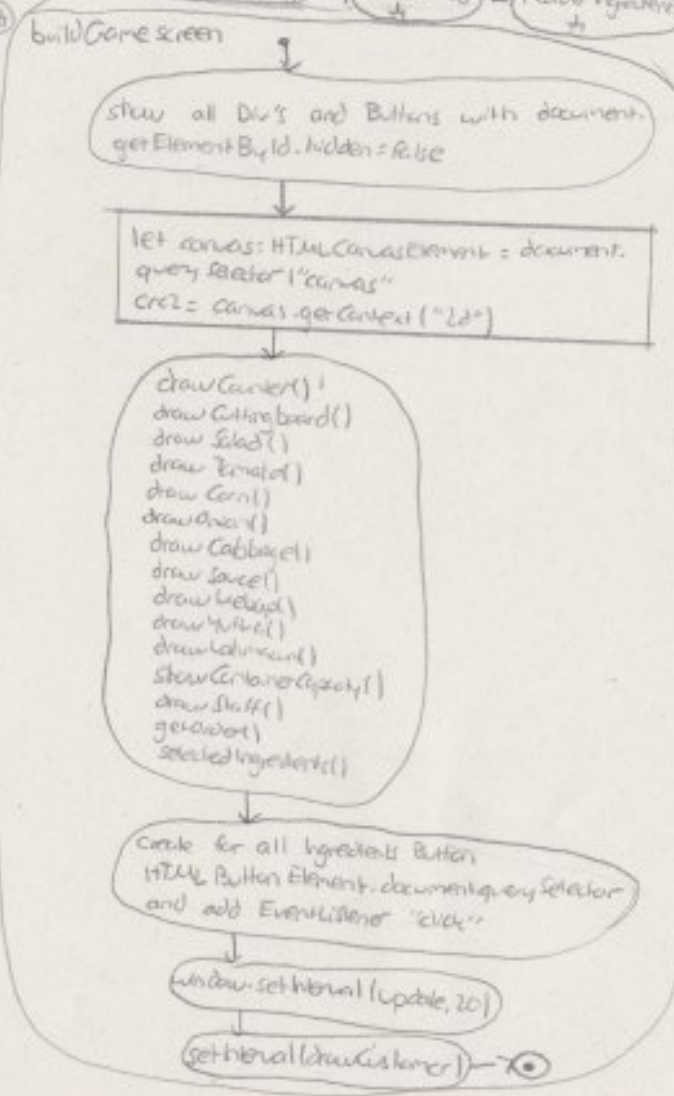
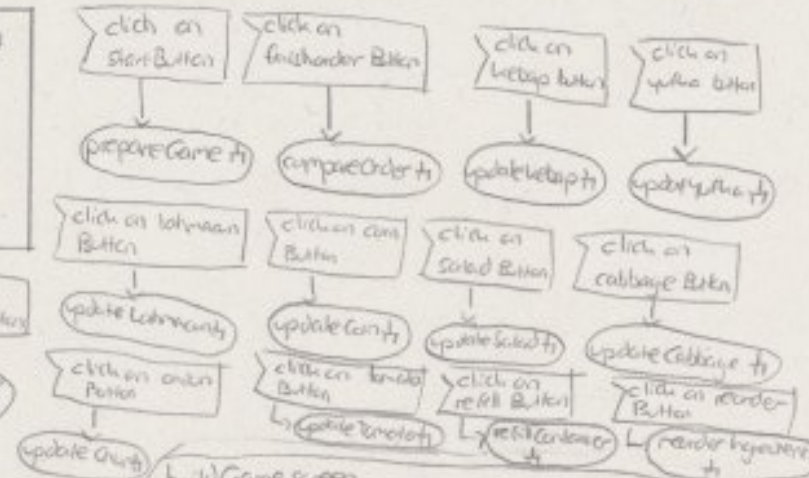
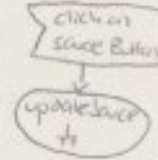
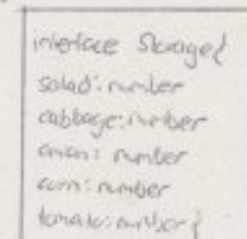
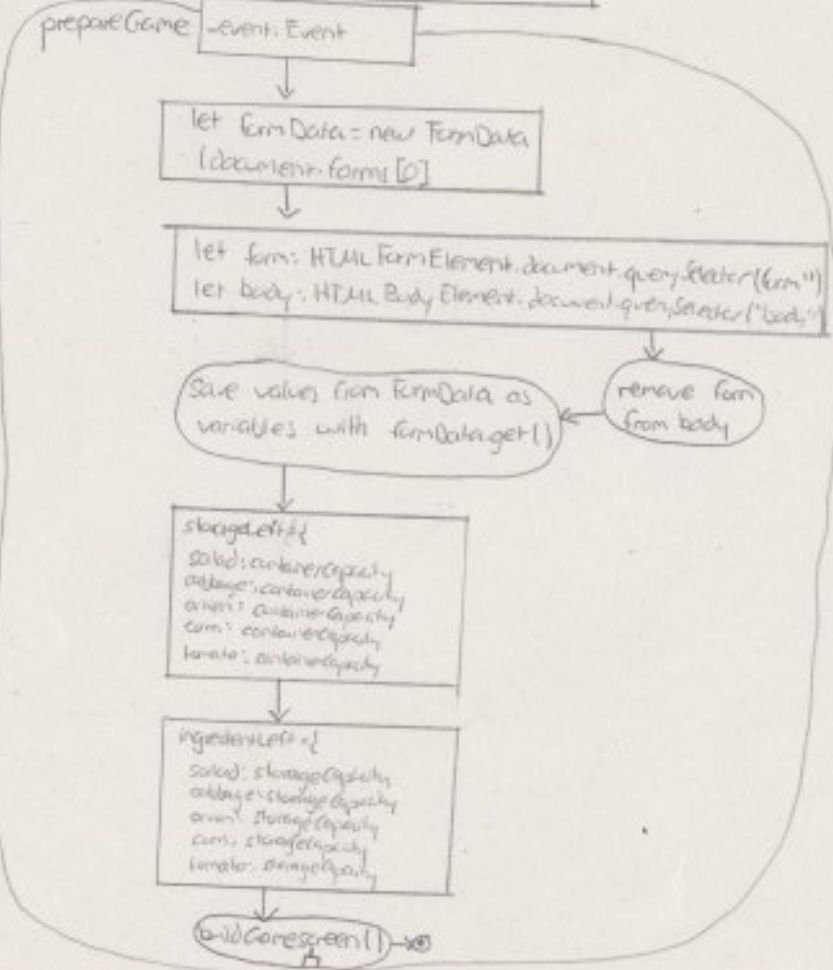
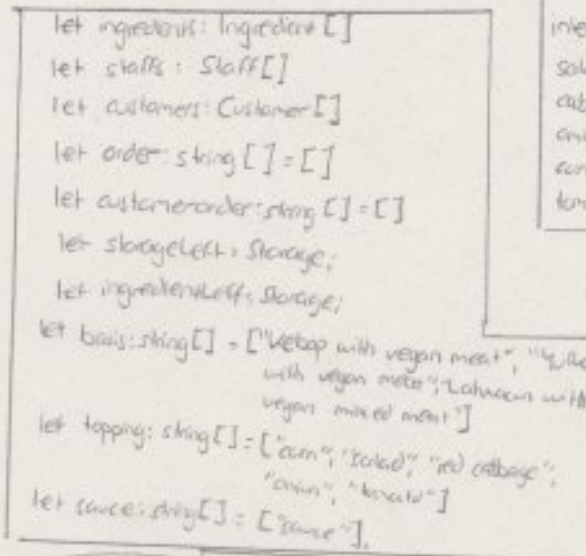
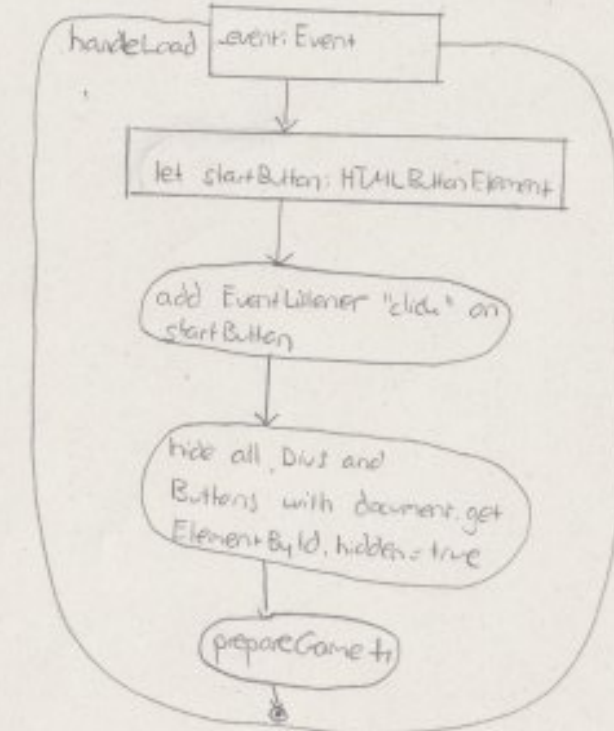
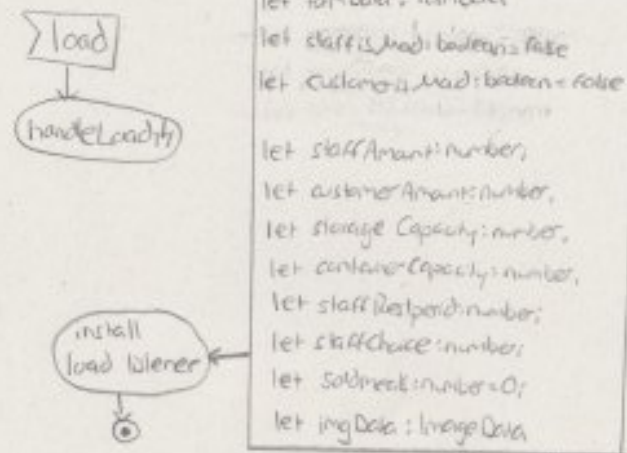
<button id="sauce"></button>

<button id="yufka"></button>

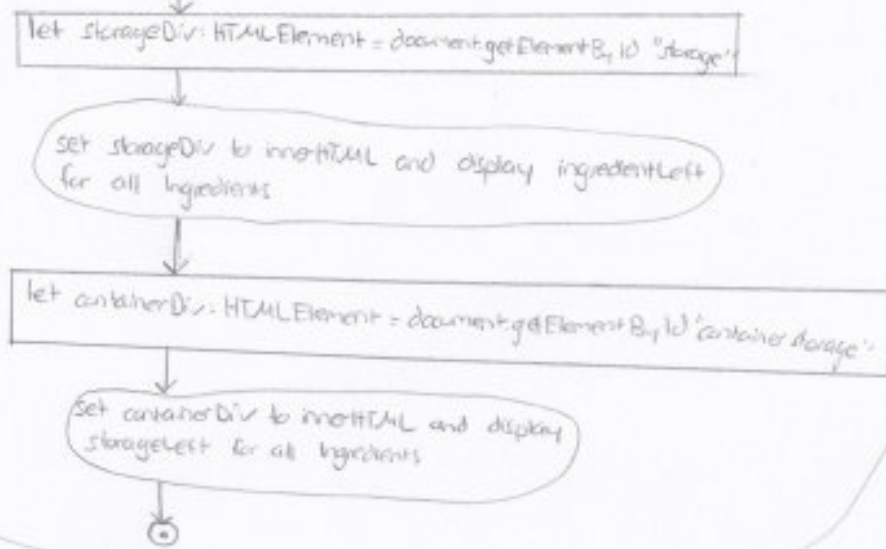
<button id="salad"></button>

Vegan Diner Trailer: Class Diagram

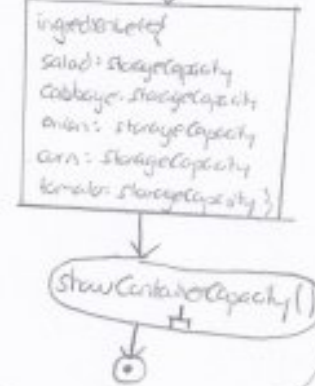




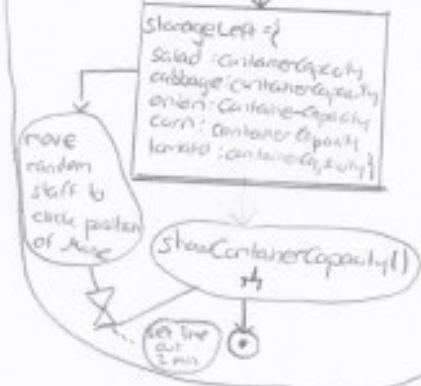
ShowContainerCapacity



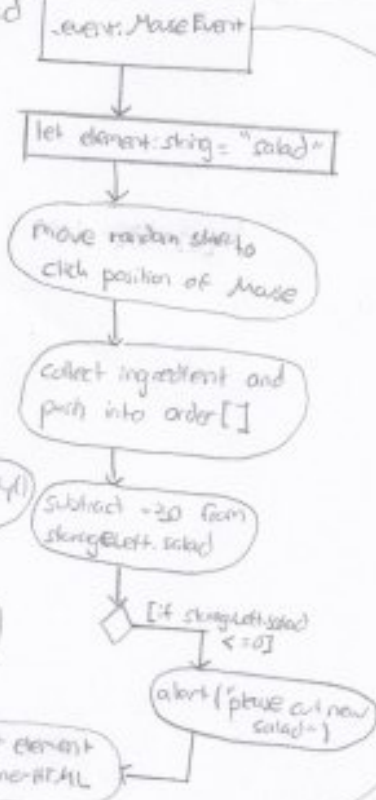
renderIngredients



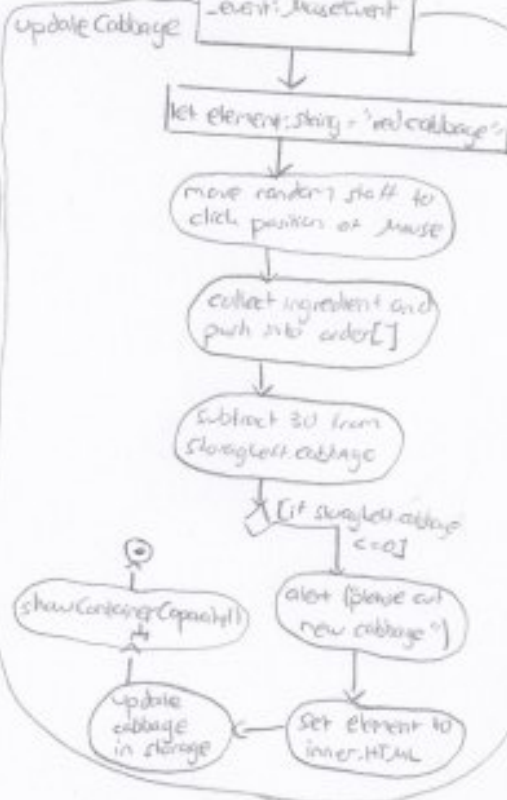
refillContainer



updateSalad



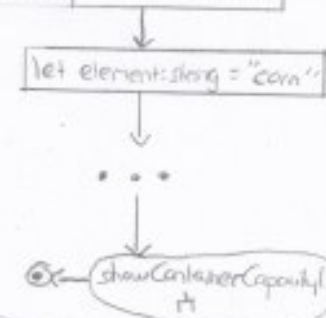
updateCabbage



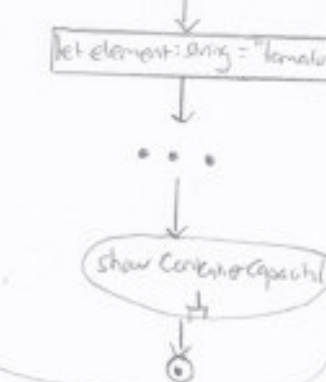
updateOnion

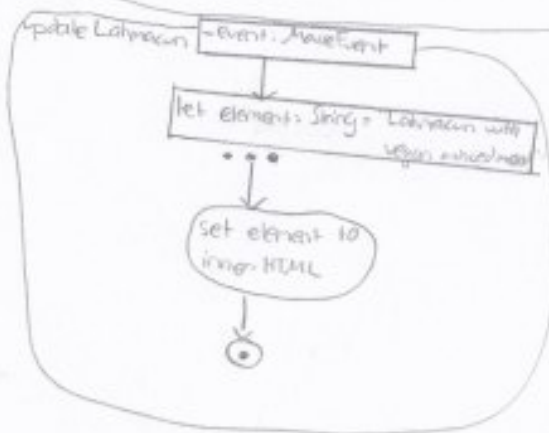
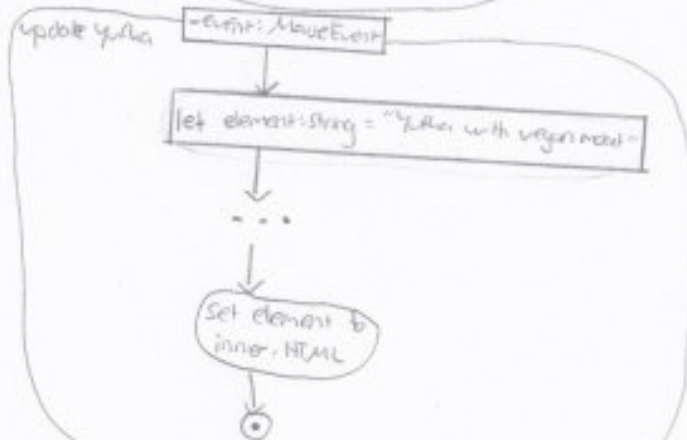
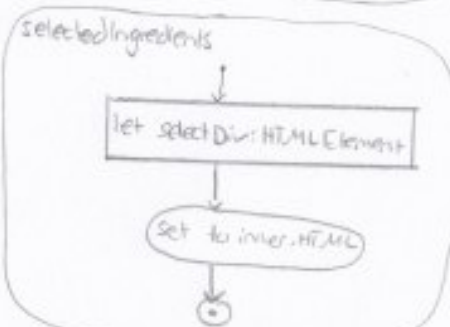
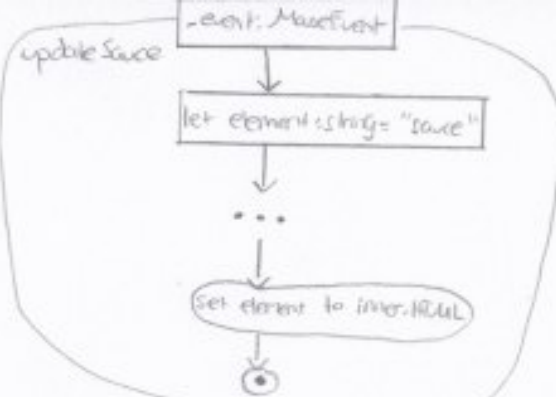
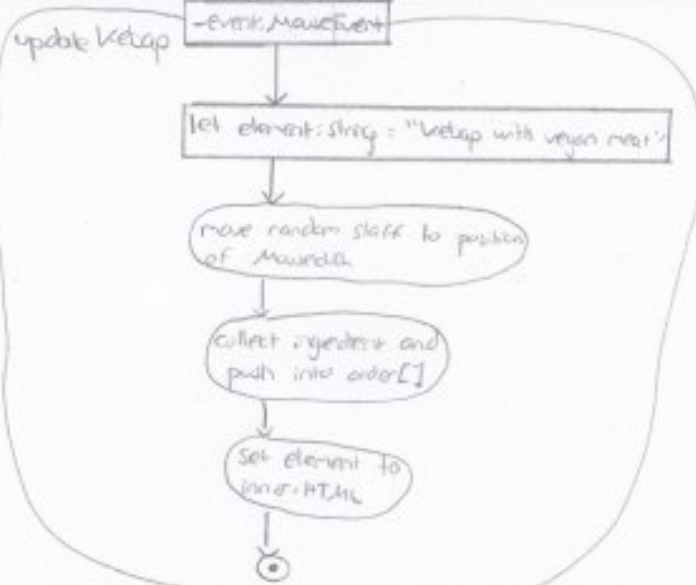


updateCorn



updateTomato





drawCustomer

let interval: number = setInterval()

create new customer

push into customer[]

customer.draw()

if customer.length == customer.length

clearInterval(interval)

customer.length = 0

setTimeout 2min

customer.stressed()

setTimeout 2min

customer.mad() and subtract 1 from overall satisfaction

set customer's mad to true and delete customer

getOrder

let object 1: number = Math.floor(Math.random() * base.length)
let object 2: number = Math.floor(Math.random() * topping.length)
let object 3: number = Math.floor(Math.random() * sauce.length)

let objects: string = base[object 1] + " " + topping[object 2] + " " + sauce[object 3]

push object into customerorder[]

let order: string[] = [base[object 1] + topping[object 2] + sauce[object 3]]

let orderDiv: HTMLElement = document.getElementById("order")

set order to innerHTML

draw Salad

create new salad

push into ingredient[]

repeat for Tomato, Corn, Onion, Red Cabbage, Sauce, Veggies, Yucca and Lacinato

ingredient.draw()

drawContainer

- position: Vector

draw container for every ingredient with fillRect()

add color with fillStyle()

drawCanter

- position: Vector

ctx2.beginPath()

drawCanter with fillRect()

add color with fillStyle

drawCuttingboard

- position: Vector

ctx2.beginPath()

drawCuttingboard with fillRect()

add color with fillStyle

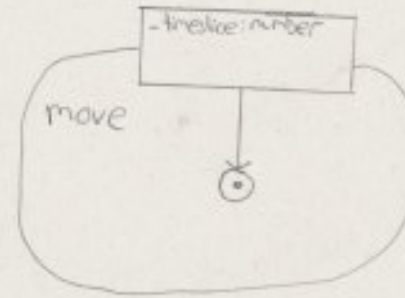
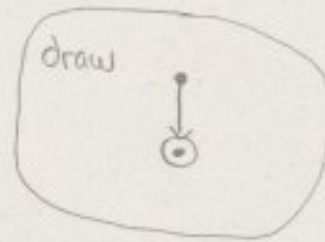
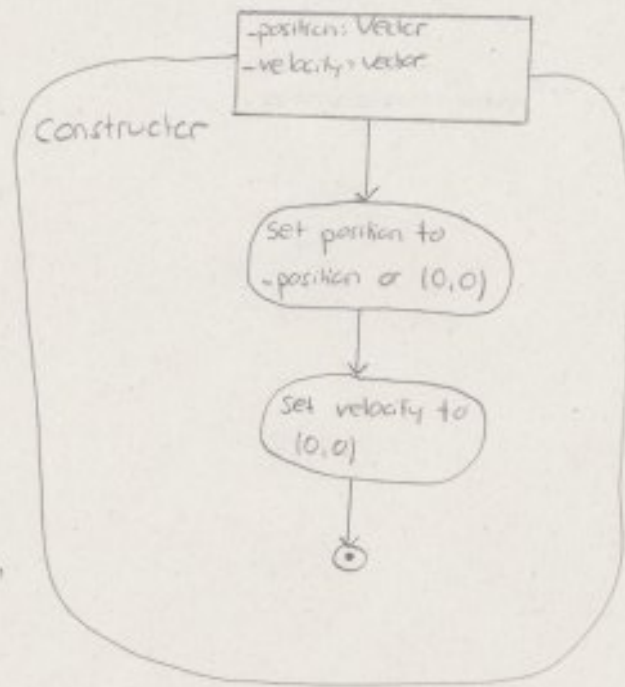
draw knife with arc 0.25 * Math.PI

add color with fillStyle

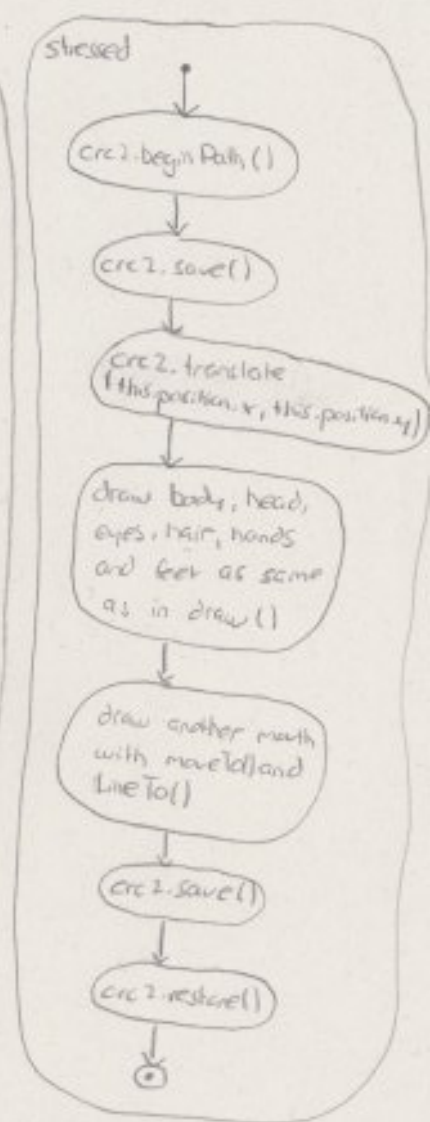
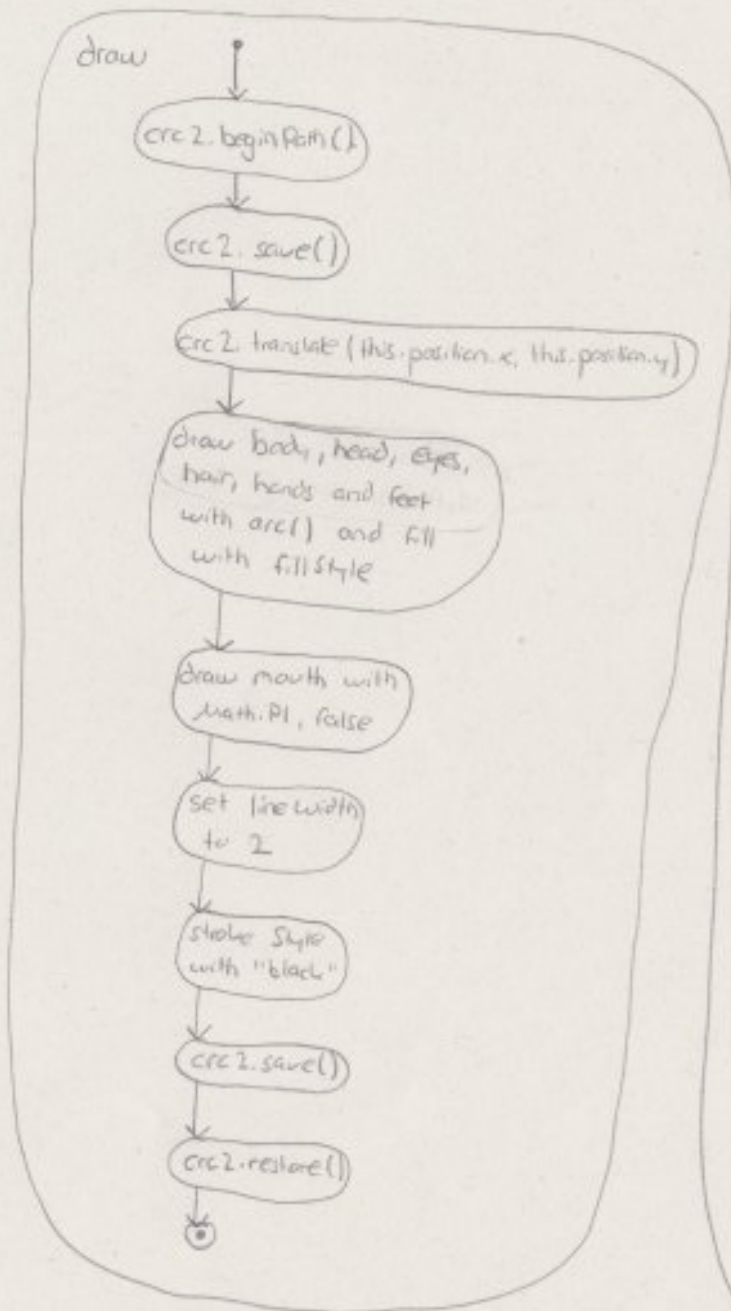
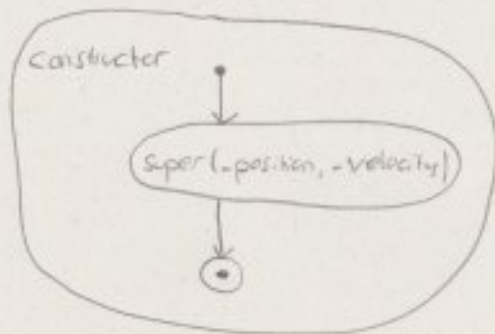
fill()

ctx2.closePath()

Vegan Döner Trainer: Activity Diagrams: Superclass Person



Vegan Diner Trainer: Activity Diagram: Subclass Customer



Vegan Döner Trainer: Activity Diagram: Subclass Staff

Constructor

super(-position, -velocity)



draw

crc2.beginPath()

crc2.save()

crc2.translate
(this.position.x, this.position.y)

draw body, head, eyes, hair,
hands and feet with arc() and fillStyle

draw mouth with
Math.PI, false

set lineWidth
to 2

stroke Style with
"black"

crc2.save()

crc2.restore()



mad

crc2.beginPath()

crc2.save()

crc2.translate
(this.position.x, this.position.y)

draw body, head, eyes,
hair, hands and feet
as same as in
draw()

draw another mouth
with Math.PI, true

set lineWidth
to 2

Stroke Style
with "black"

crc2.save()

crc2.restore()



sleep

crc2.beginPath()

crc2.save()

crc2.translate
(this.position.x, this.position.y)

draw body, head, hair and
feet same as in draw()

draw another mouth
with begin Path() and
moveTo()

draw other eyes with
7 * Math.PI

crc2.save()

crc2.restore()



move

- position: Vector
- velocity: Vector
- timestep: number

let offset: Vector = this.velocity.copy()

offset.scale(-timestep)

this.position.add(offset)

if this.position.x < 0

this.position.x +=
crc2.canvas

if this.position.y < 0

this.position.y +=
crc2.canvas

if this.position.x > canvas.width

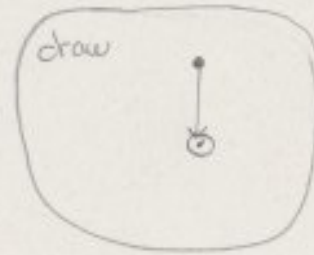
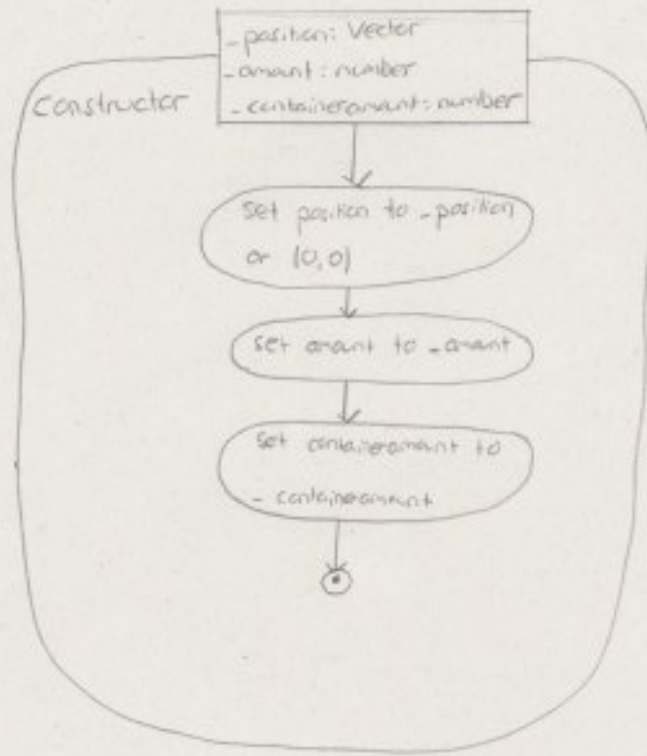
this.position.x -=
canvas.width

if this.position.y > canvas.height

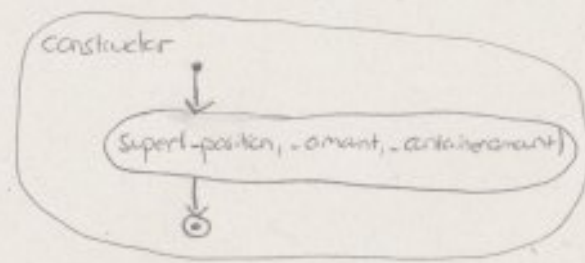
this.position.y -=
canvas.height



Vegan Döner Trainer: Activity Diagram: Superclass Ingredient

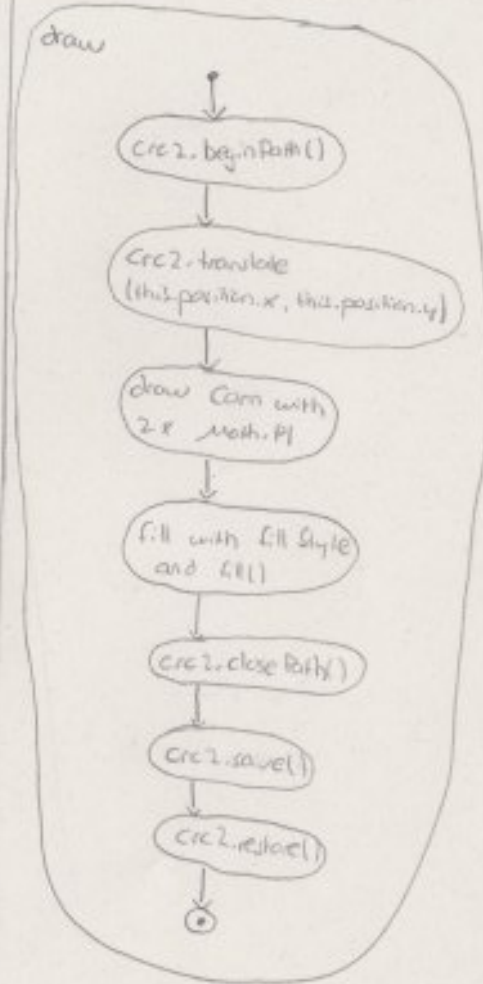
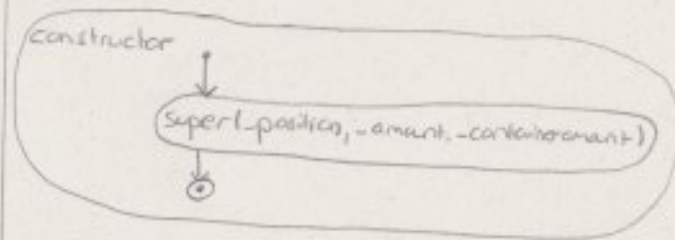


Vegan Diner Trainer: Activity Diagrams: Subclass: Cabbage

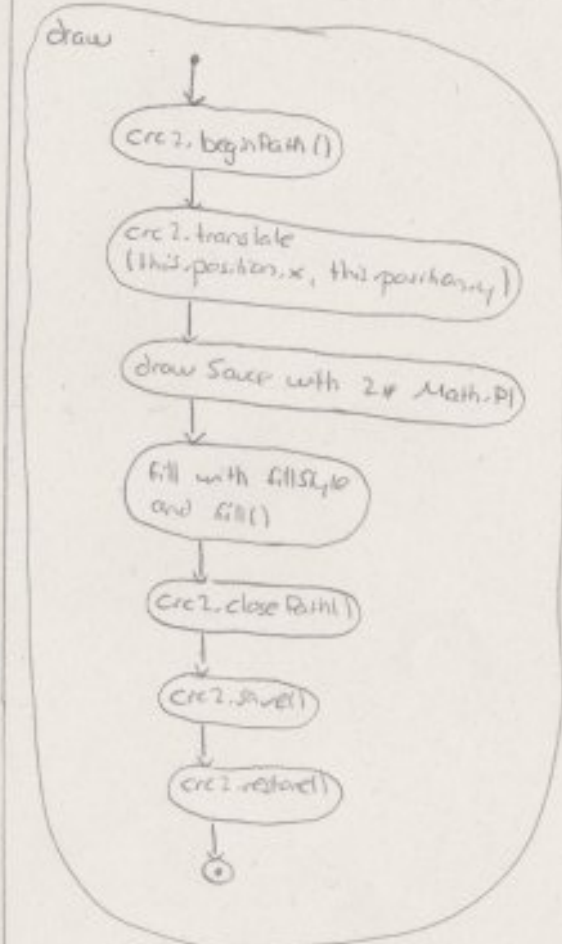
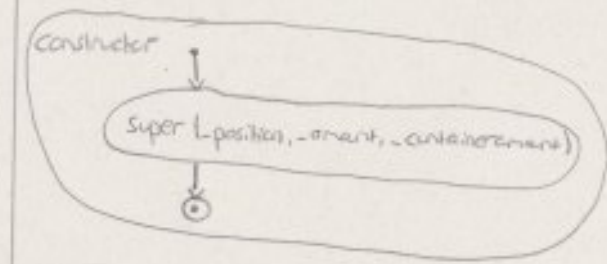


for drawing a circle
beginPath
draw circle
closePath
fillStyle

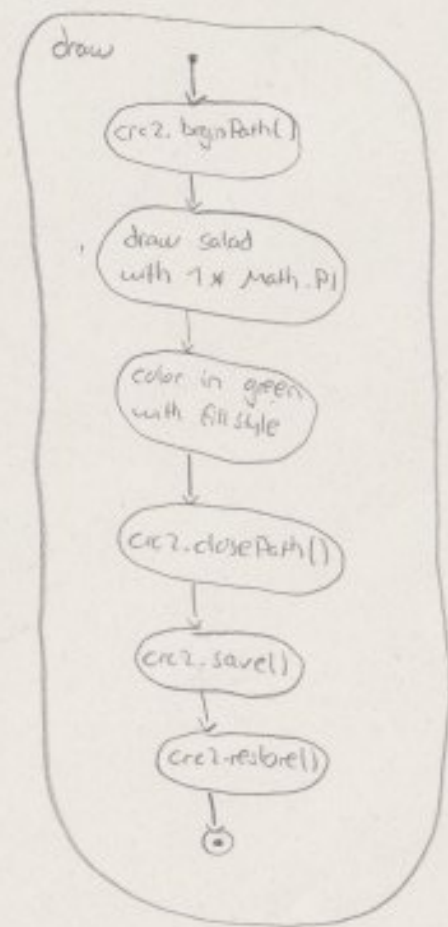
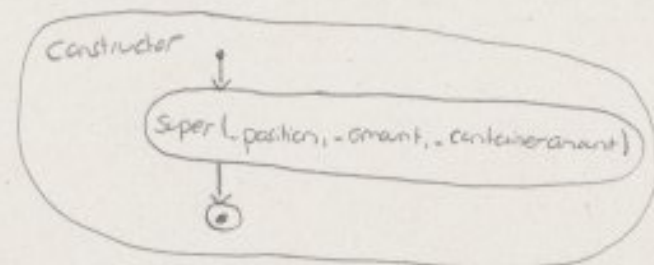
Subclass Corn



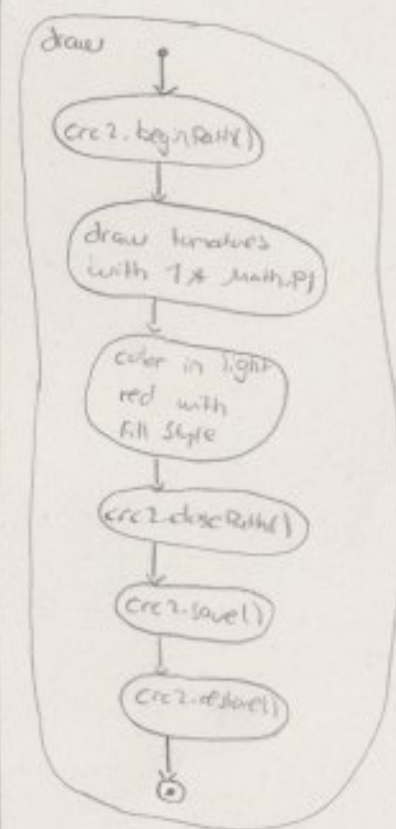
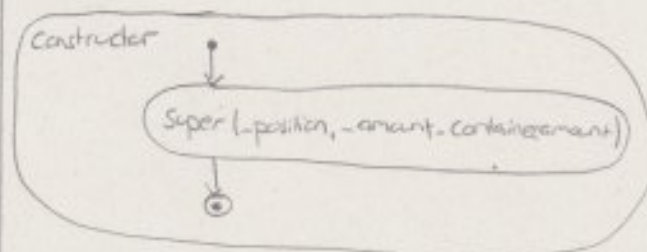
Subclass Sauce



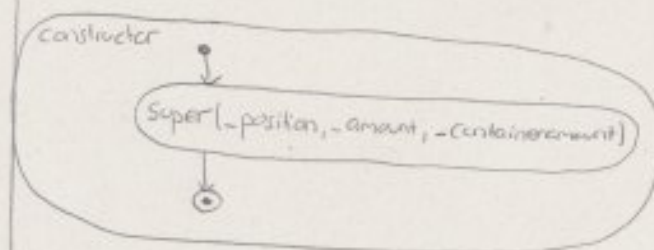
Vegan Döner Trainer: Activity Diagram: Subclass Salad



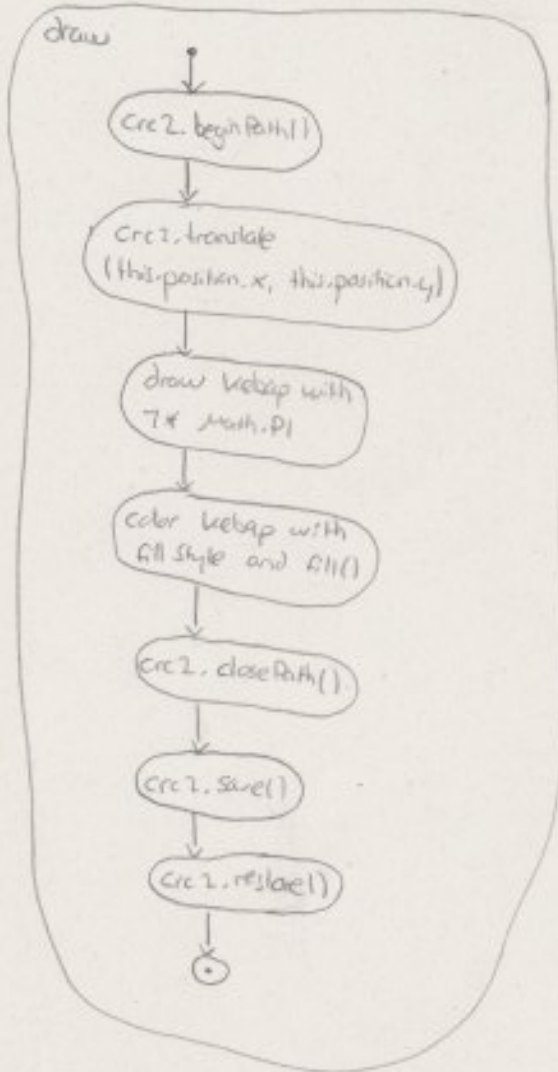
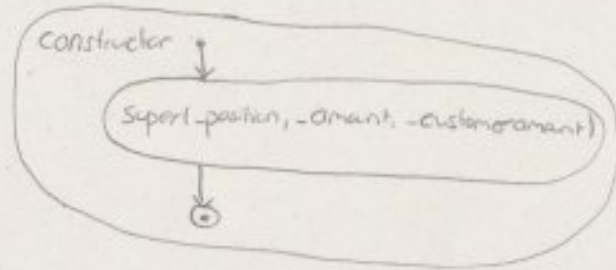
Subclass Tomato



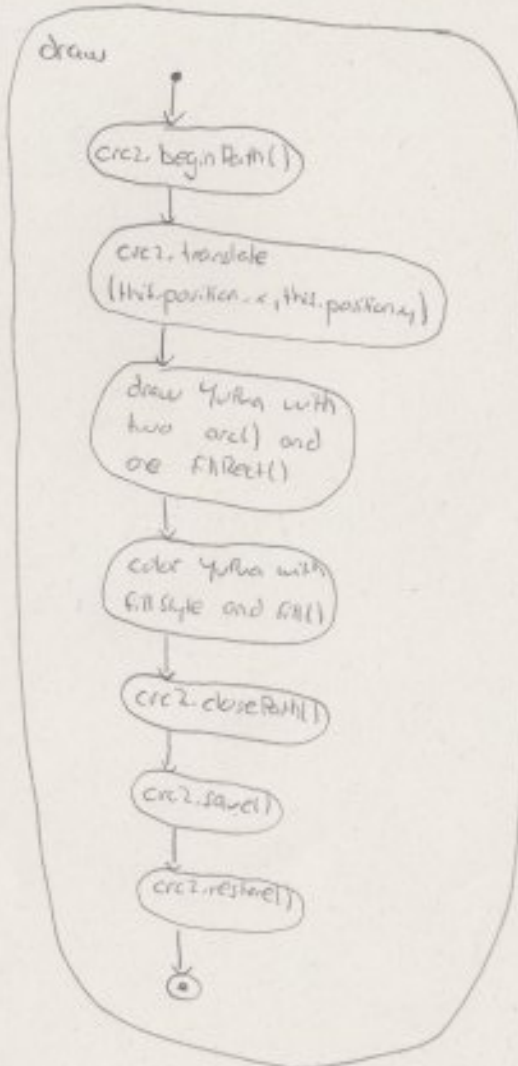
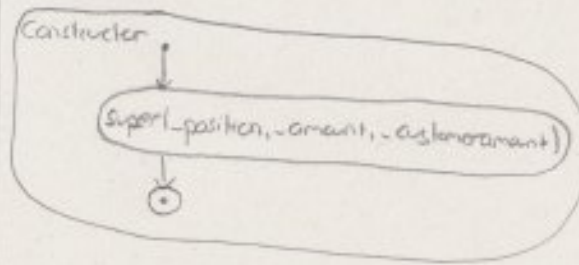
Subclass Onion



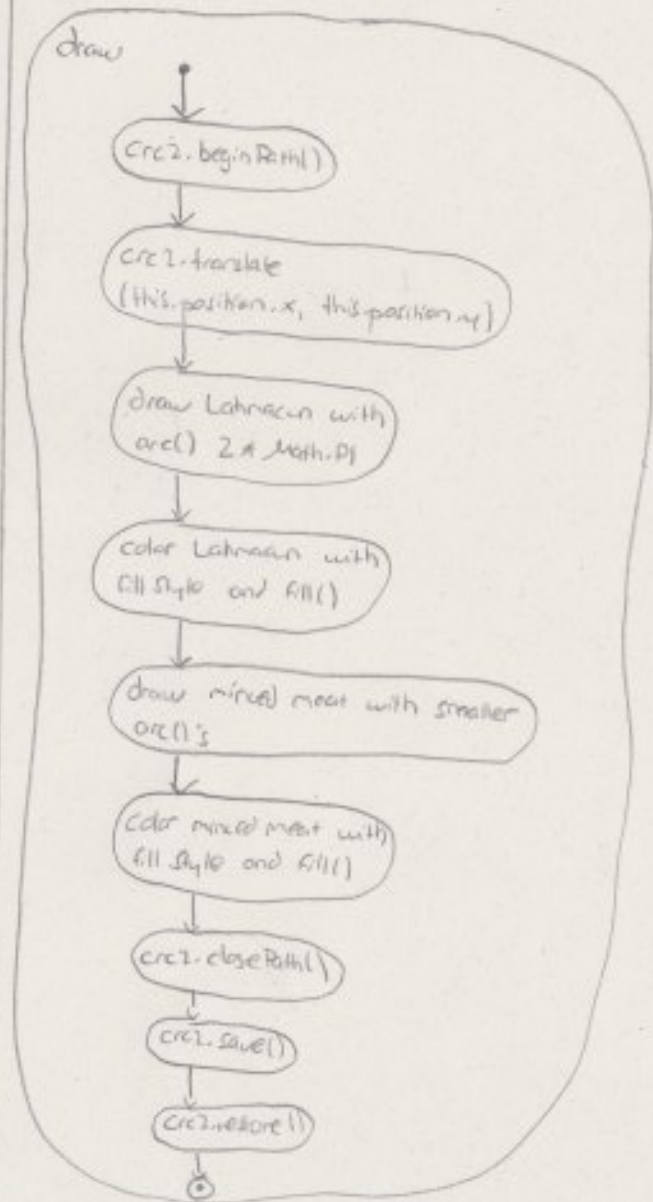
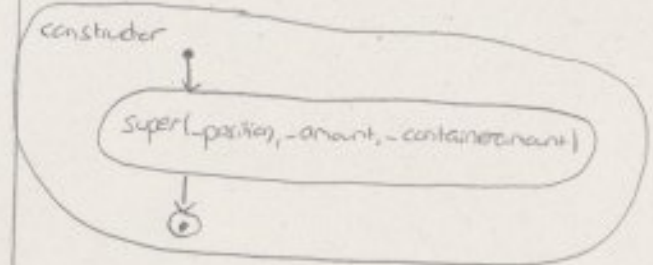
Vegan Diner Trainer: Activity Diagram: Subclass: Kebap



Subclass Yufra



Subclass Lahmacun



Anwendungsdurchlauf Vegan Döner-Trainer

1. Die Anwendung wird am Computer geöffnet
2. Die Einstellungen im Settings-Formular werden ausgewählt:
 - a. staff amount
 - b. customer amount per 1 minute
 - c. storage capacity
 - d. container capacity
 - e. staff rest period
3. Auf den „start Game“ Button klicken → Das Spiel öffnet sich
4. Kunden kommen automatisch und teilen Ihre Bestellungen mit
5. Klicke auf die benötigten Zutaten, um diese einzusammeln. Ein zufällig gewählter Mitarbeiter läuft dann zu den Zutaten und sammelt diese ein, dabei wird die Anzahl der Zutaten in den einzelnen Containern, sowie im Storage weniger.
6. Nach 2 Minuten wird der erste Mitarbeiter müde und nach weiteren zwei Minuten sauer, wenn er sauer geworden ist, geht er automatisch für 2 Minuten in Pause und arbeitet nicht mehr.

7. Wenn alle Zutaten richtig eingesammelt wurden, wird auf den „finish order“ Button geklickt um die Bestellung mit den eingesammelten Zutaten zu vergleichen.
8. Wenn die Bestellung richtig übergeben wurde, erhält die “overall satisfaction“ Anzeige, sowie die „sold meals“ Anzeige eine +1, wenn nicht ein -1.
Wenn der Kunde zu lange warten muss, wird er nach 2 Minuten erst ungeduldig und nach weiteren 2 Minuten sauer. Wenn der Kunde sauer geworden ist, verlässt er den Döner-Laden und die Bestellung verfällt.
9. Wenn einige Bestellungen abgearbeitet wurden, leeren sich die Container, sowie der Storage. Nun erscheint eine Aufforderung, die Container wieder aufzufüllen. Dafür wird auf den „cuttingboard“ Button geklickt. Ein zufällig ausgewählter Mitarbeiter läuft zum Cuttingboard und schneidet dort für zwei Minuten. Dadurch werden die Container wieder gefüllt. Ist der Storage leer, kann dieser durch den „reorder“ Button wieder gefüllt werden.