

Konzept – erste Überlegungen

Der Nutzer spielt an sich zwei verschiedene Spiele. Ein Spiel, in dem er verschiedene Eissorten selbst zusammenstellen kann, und dann ein anderes, in dem er diese verkauft. Die einzige feste Verbindung zwischen den beiden Spielen sind die festgelegten Eissorten. Diese müssen auf einem Server gespeichert sein, damit das Spiel wie vorgegeben auf mehreren Geräten mit dem gleichen Eis gespielt werden kann. Die beiden Teile des Spiels können durch eine Seite (quasi ein Startbildschirm) verbunden werden, in dem es auch die Möglichkeit gibt, das Spiel zurückzusetzen.

Eis zusammenstellen

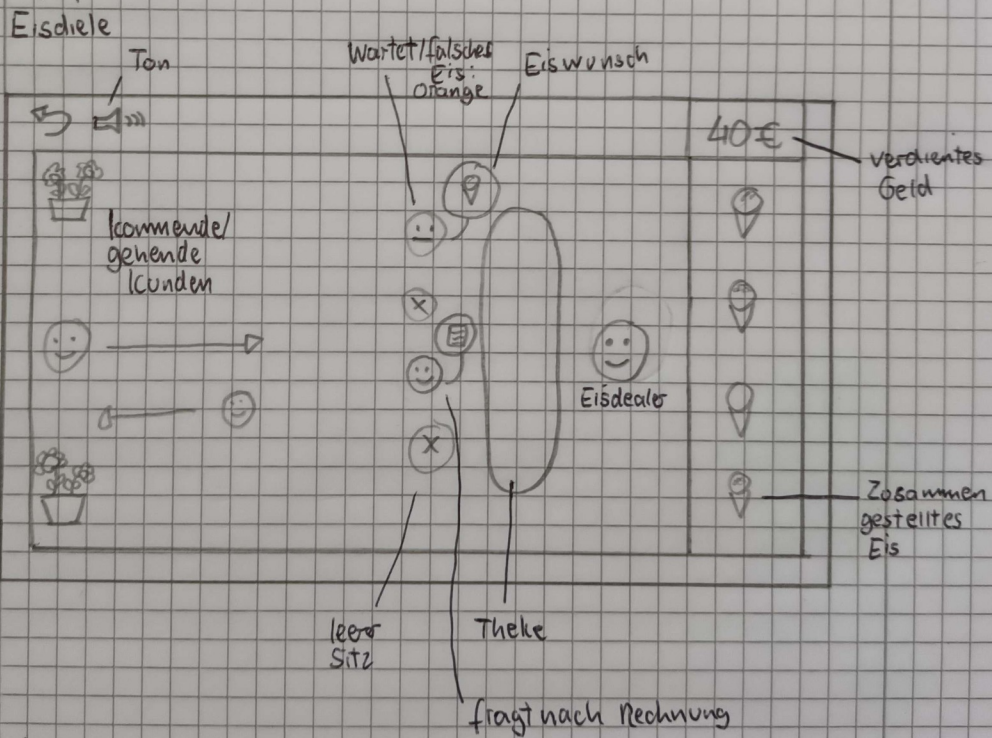
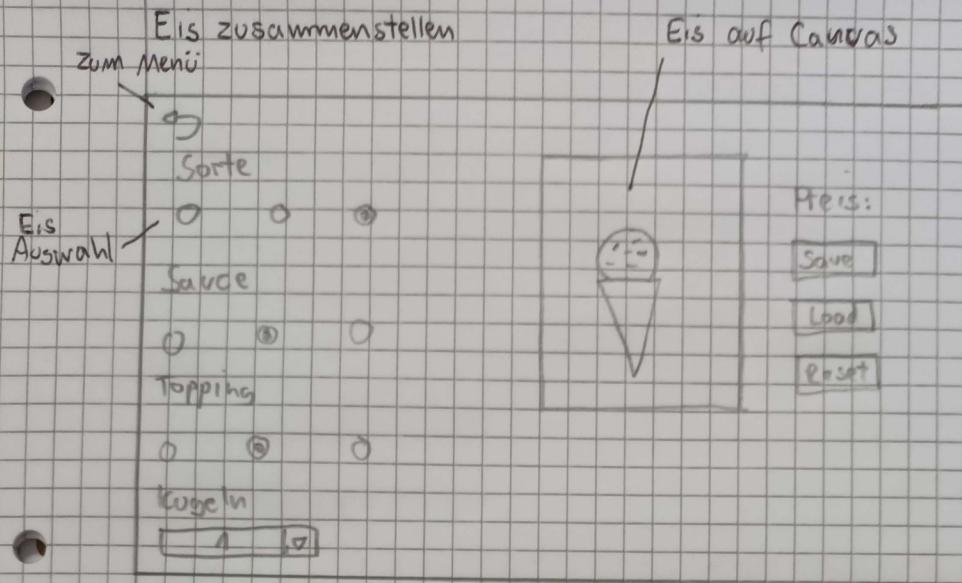
- Auswahl der verschiedenen Eissorten, Toppings usw durch Auswahlmöglichkeiten wie Radiobuttons oder sogar drag&drop
- Grafische Darstellung des zusammengestellten Eis auf canvas
- Möglichkeiten zum speichern (Eis rutscht in Sortiment für Eisdiele), laden (bereits erstelltes Eis kann verändert werden), zurücksetzen (wenn man von vorne beginnen will)

Eisdiele

- Nutzer hat seine verschiedenen Eissorten hinter der Theke. Kunden fragen nach einer der zusammengestellten Eissorten (Sprechblase?), der Nutzer kann diese per drag&drop oder per Klick ausgeben
- Nach dem der Kunde das Eis gegessen hat (Animation?), verlangt er nach der Rechnung. Durch einen Klick auf diese wird sie bezahlt. Der Kunde verlässt die Eisdiele, die Kassenanzeige wird um den preis des Eises erhöht
- Der Kunde zeigt durch Farbe seine Stimmung an (gelb orange rot). Gibt man ihm das falsche Eis oder lässt ihn länger als 10 Sekunden warten, sinkt die Stimmung um eine Stufe. Bei rot verlässt er die Eisdiele, ohne zu zahlen. Orange Stimmung steigt wieder auf Gelb, wenn sie das richtige Eis erhalten
- Kunden kommen in zufälligen Zeitabständen in die Diele. Sind alle Plätze (4? 6? 8?) belegt, verlassen sie die Diele wieder

Ich entscheide mich dafür, das Spiel für PC zu erstellen und zu optimieren. Grafik und Auflösung sind auf diesem Gerät besser und, viel wichtiger, ziemlich einheitlich. Die Größen von Smartphone-Displays variieren und machen die Erstellung eines räumlichen Spiels deutlich schwieriger, da alle Elemente so verteilt werden müssten, das es auf verschiedenen Displaygrößen gut bedienbar ist. Außerdem sind die Elemente auf einem großen Bildschirm besser erkennbar. Dazu kommt, das PCs einfach leistungstärker sind und das Spiel so schneller geladen/mit weniger Problemen gespielt werden kann.

Skizzen



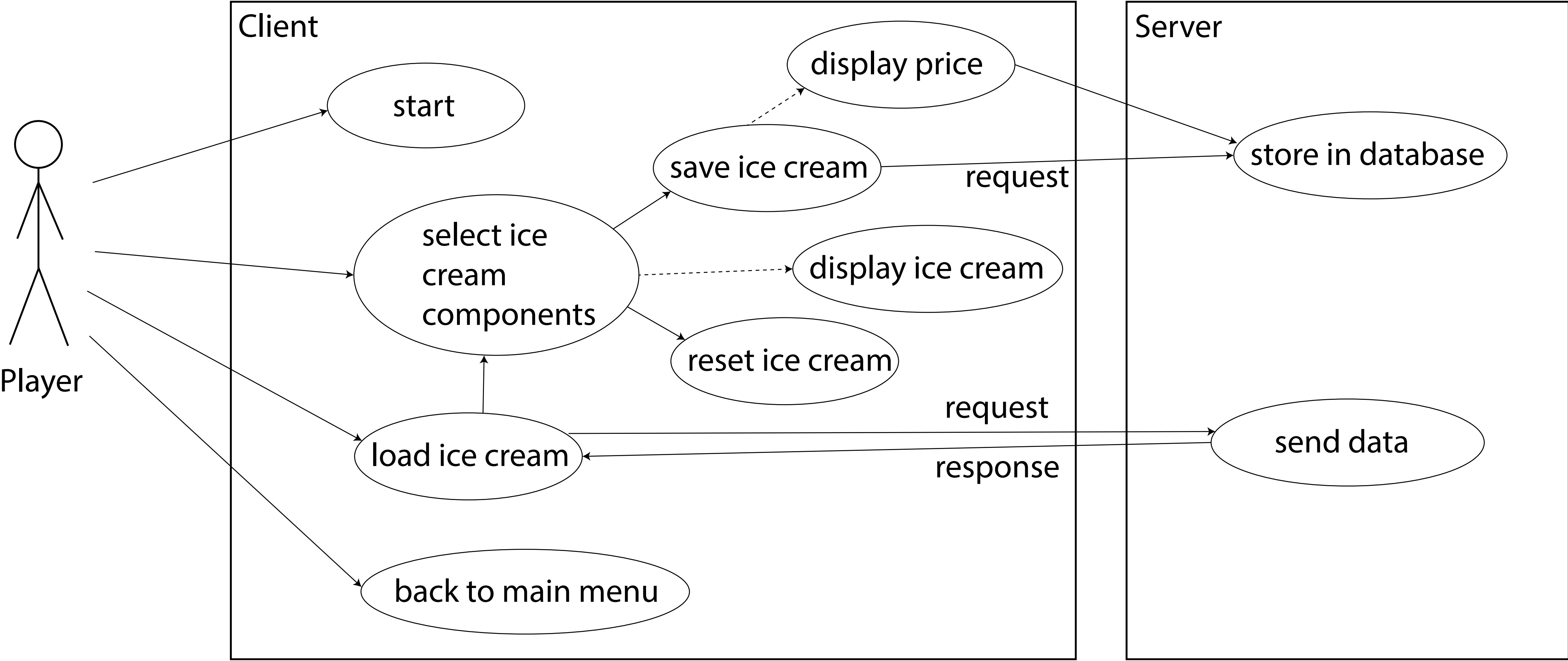
Ton: Hintergrund: Musik, Leute reden

Kunde kommt herein: Klingel

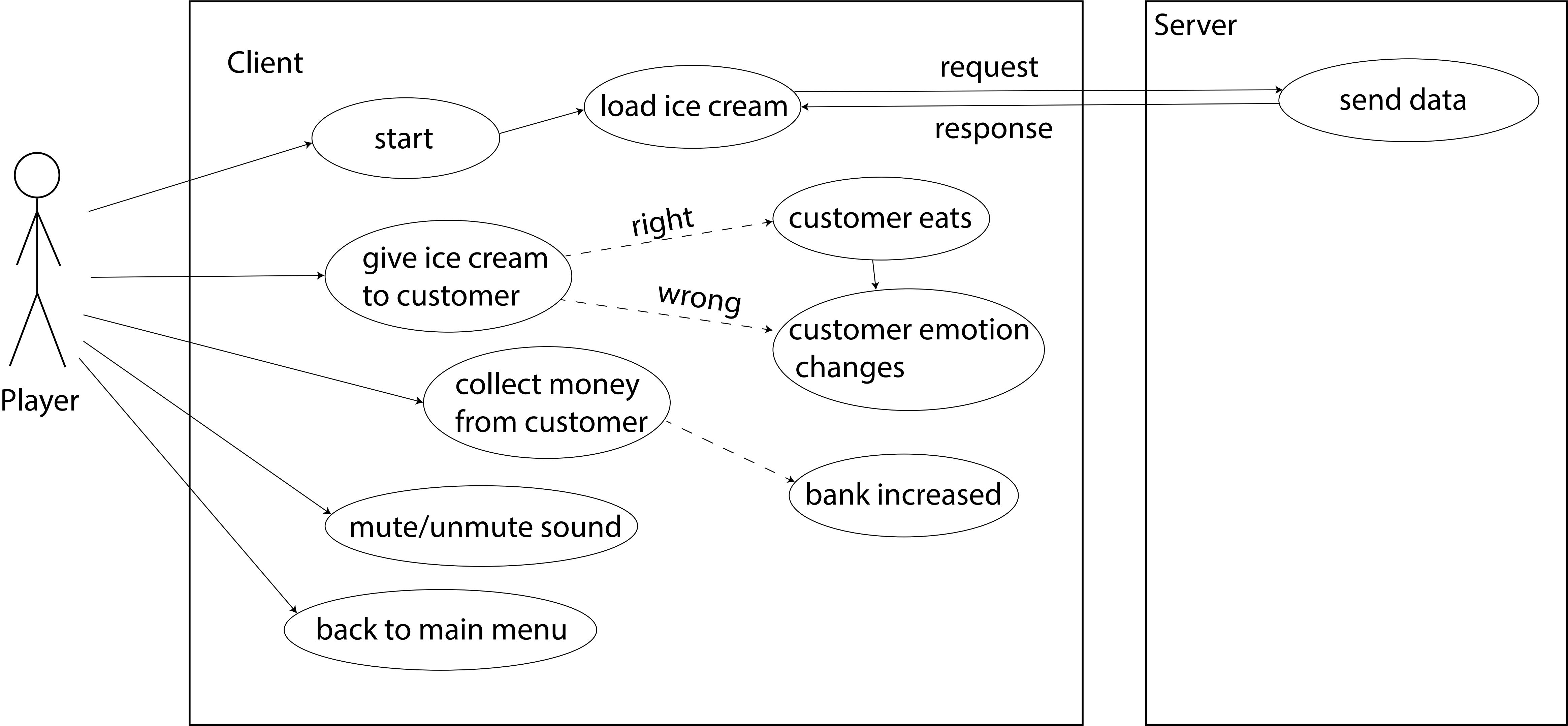
Kunde fragt nach Eis/Rechnung: "Plopp"

Kunde bezahlt: Kassengeräusch

Use-Case-Diagram: Ice Cream Creation



Use-Case-Diagram: Iceddealer



Eisdealer UI

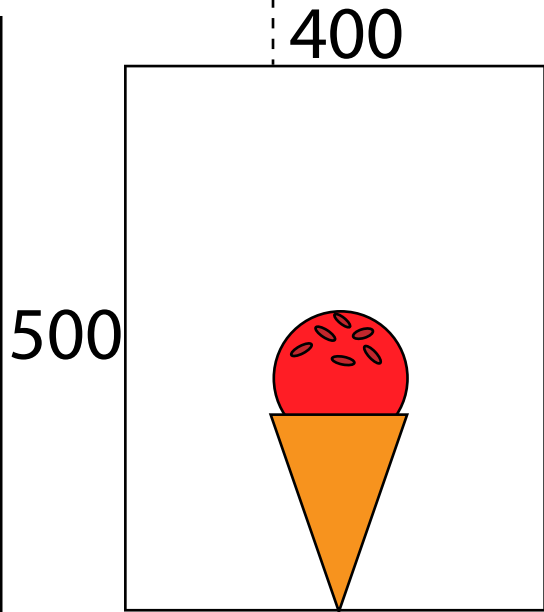
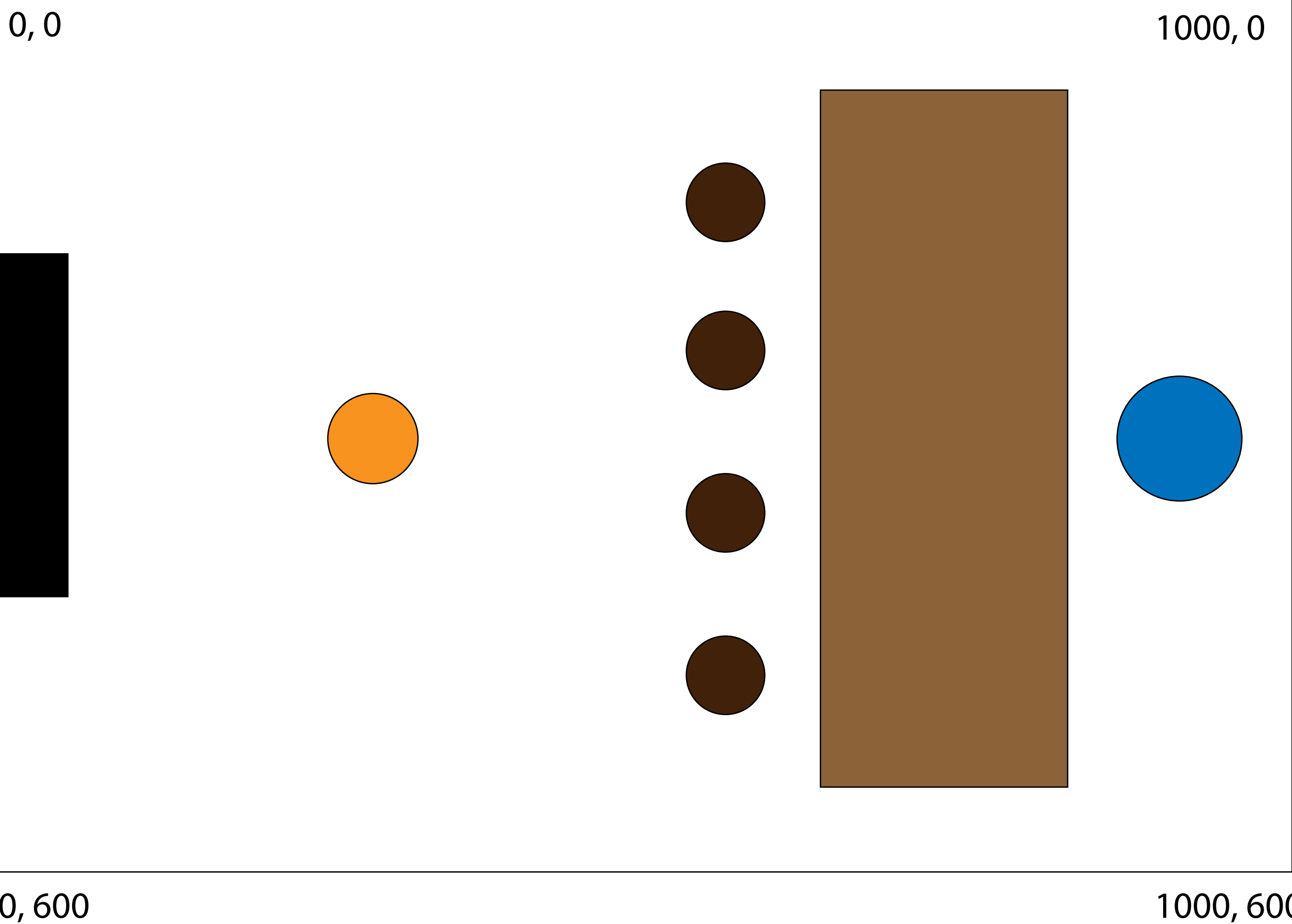
```
<h1>
```

```
<canvas>  
id = gameCanvas
```

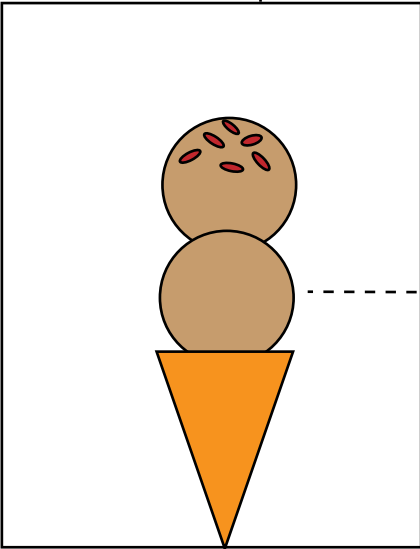
```
<canvas>  
class = iceCreamCanvas  
id = iceCreamCanvas1
```

```
<canvas>  
class = iceCreamCanvas  
id = iceCreamCanvas2
```

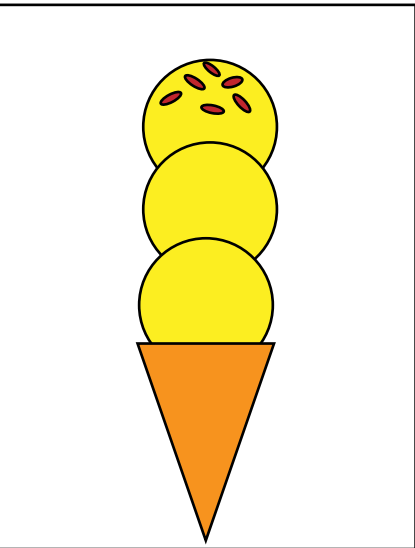
Eisdealer



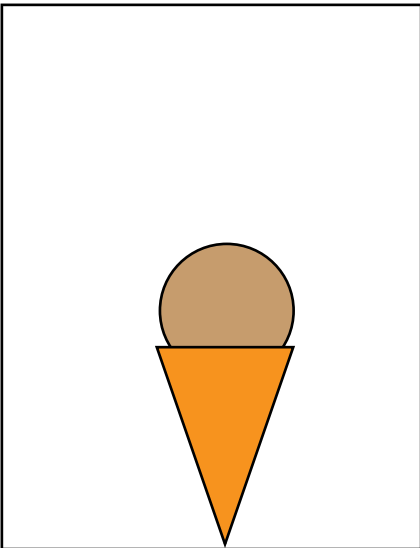
Preis: 2 Euro



Preis: 2 Euro



Preis: 2 Euro



Preis: 2 Euro

iceCream von
Eis zusammenstellen

Eisdealer:
900, 300
color: blue

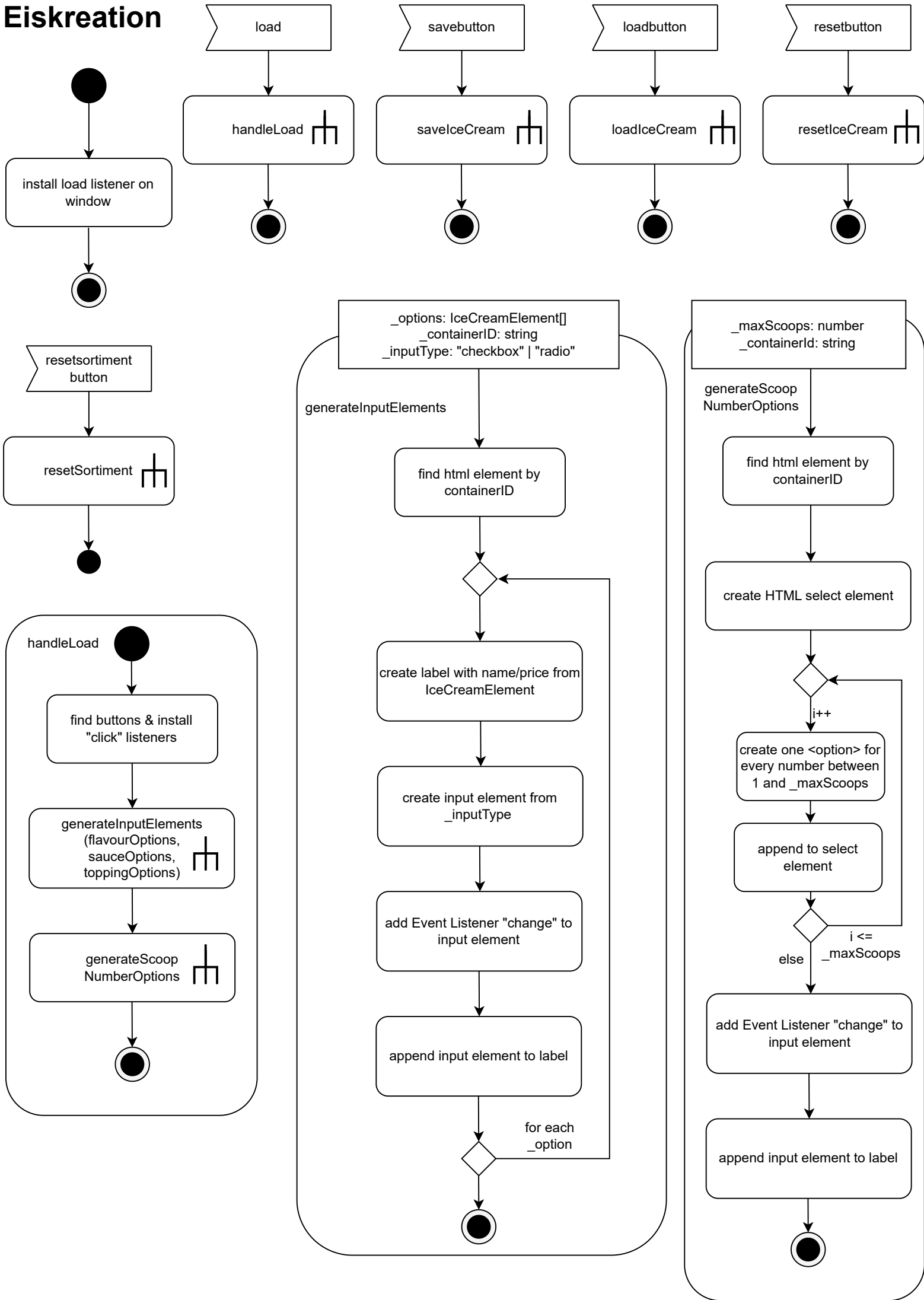
Bar:
690, 50
color: #8B4513

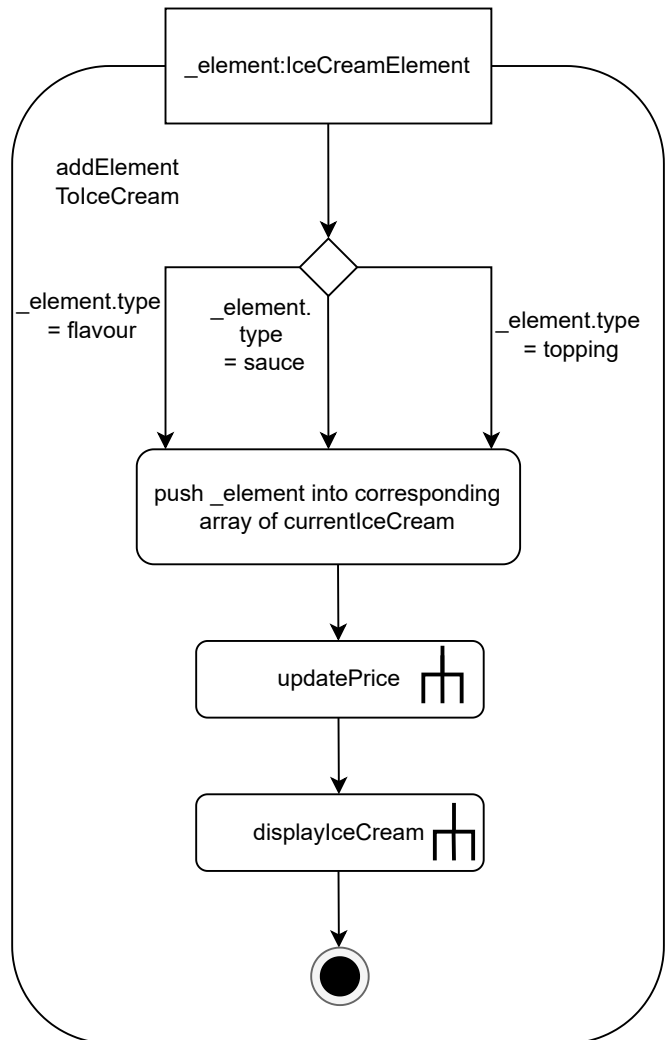
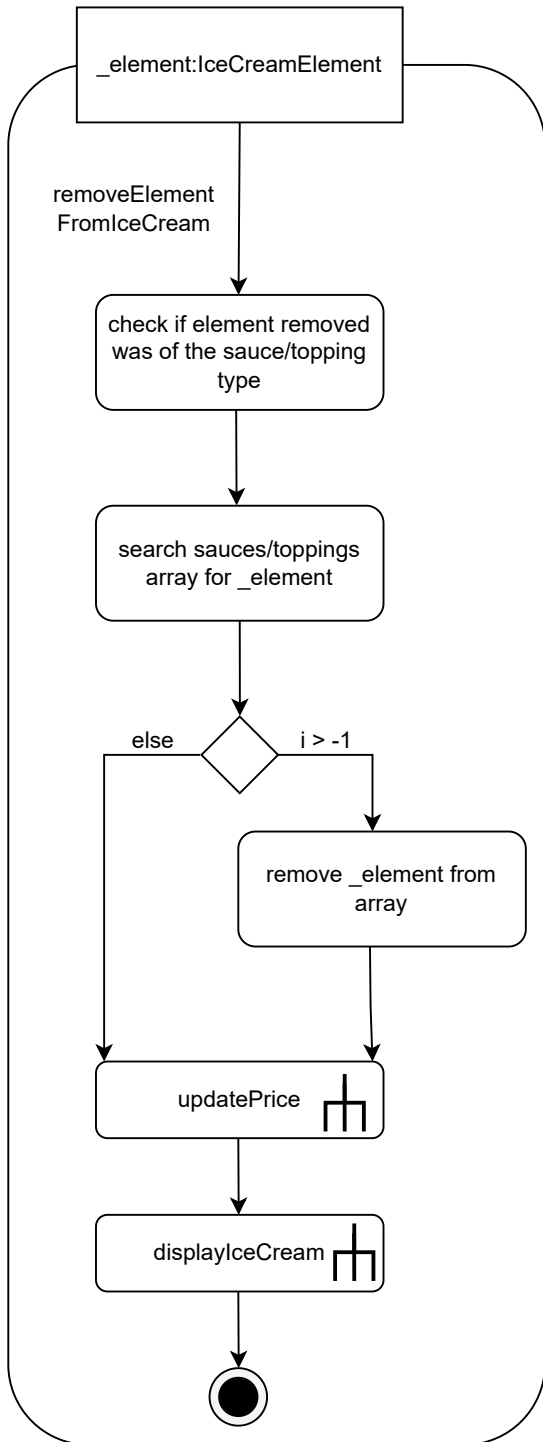
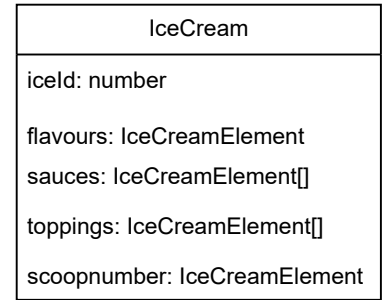
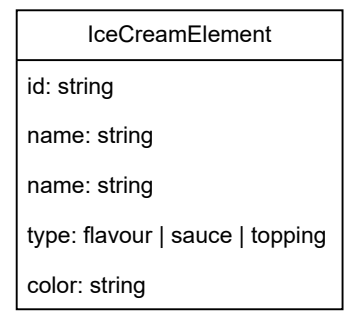
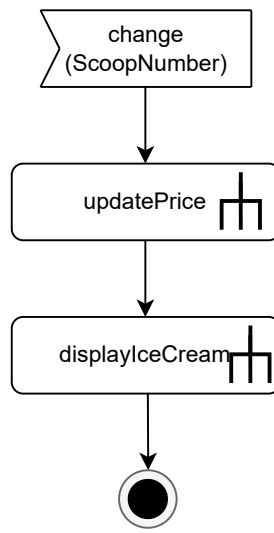
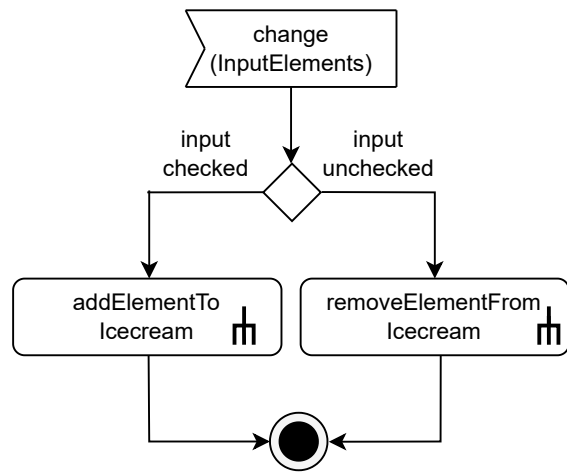
Eingang:
0, 250
color: #000000

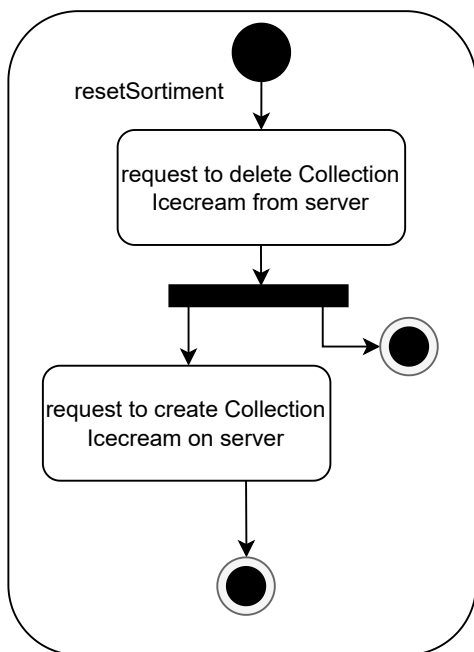
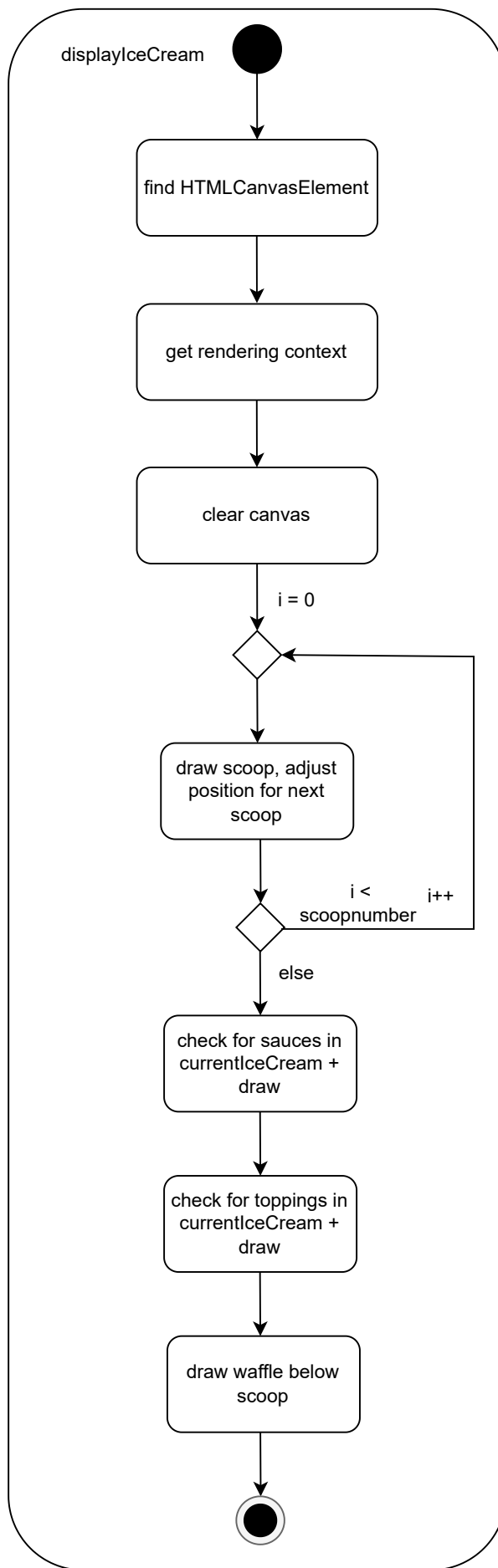
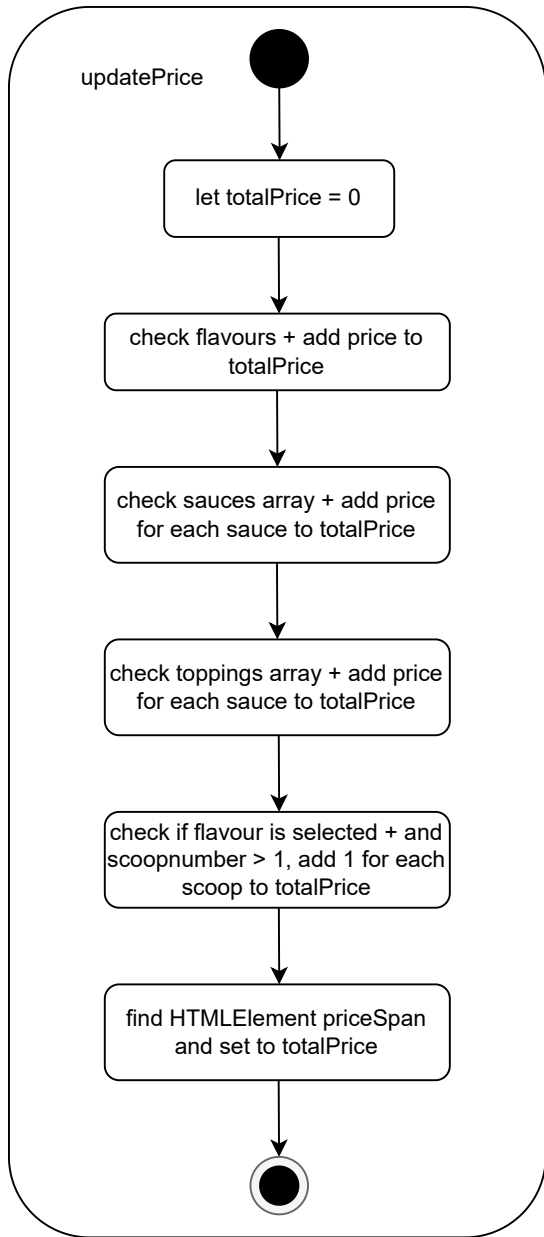
```
<canvas>  
class = iceCreamCanvas  
id = iceCreamCanvas3
```

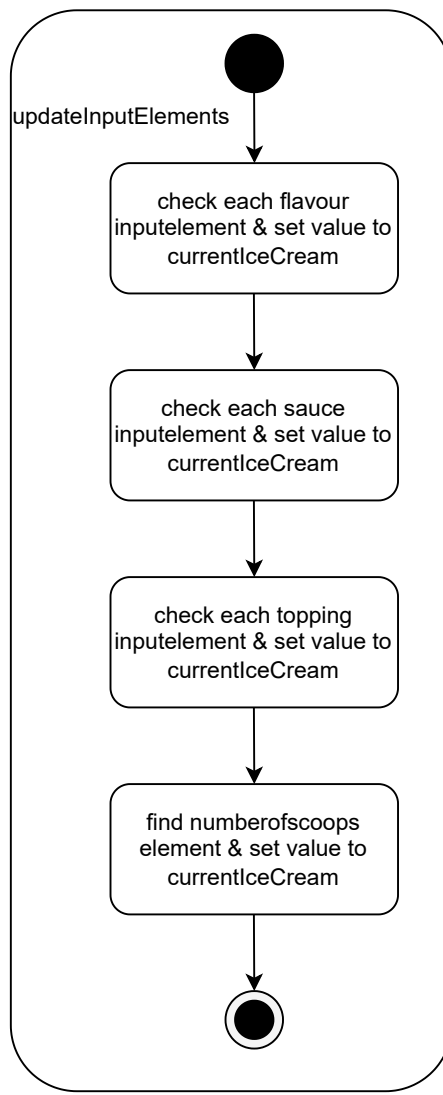
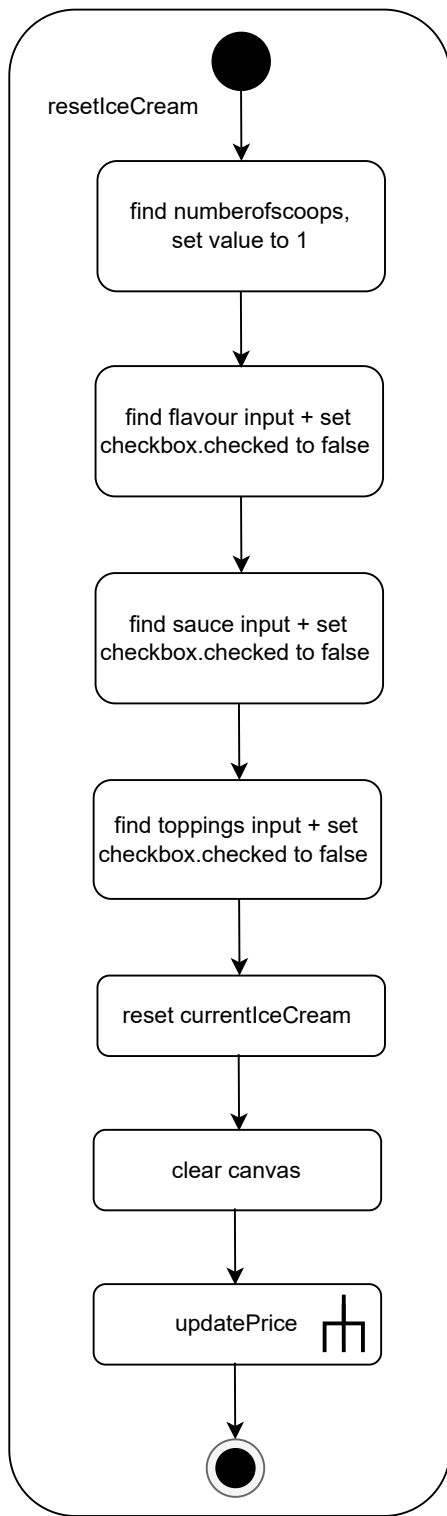
```
<canvas>  
class = iceCreamCanvas  
id = iceCreamCanvas4
```

Eiskreation

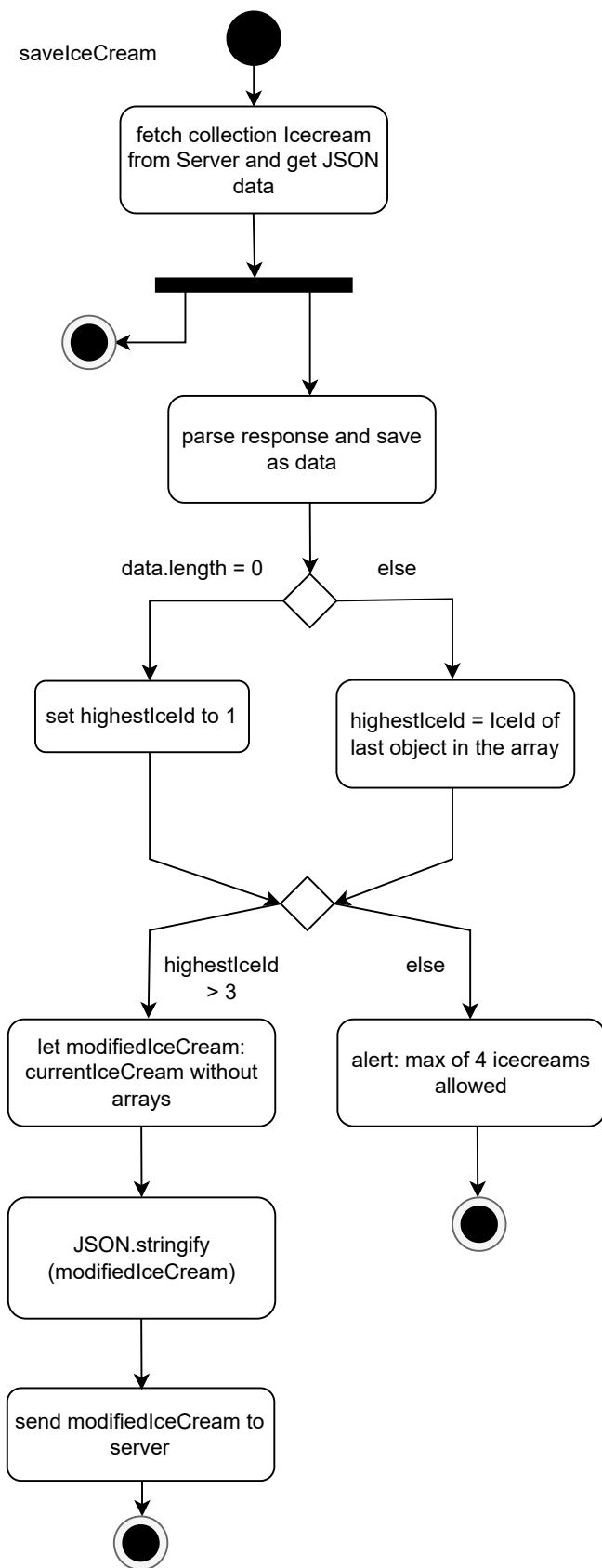




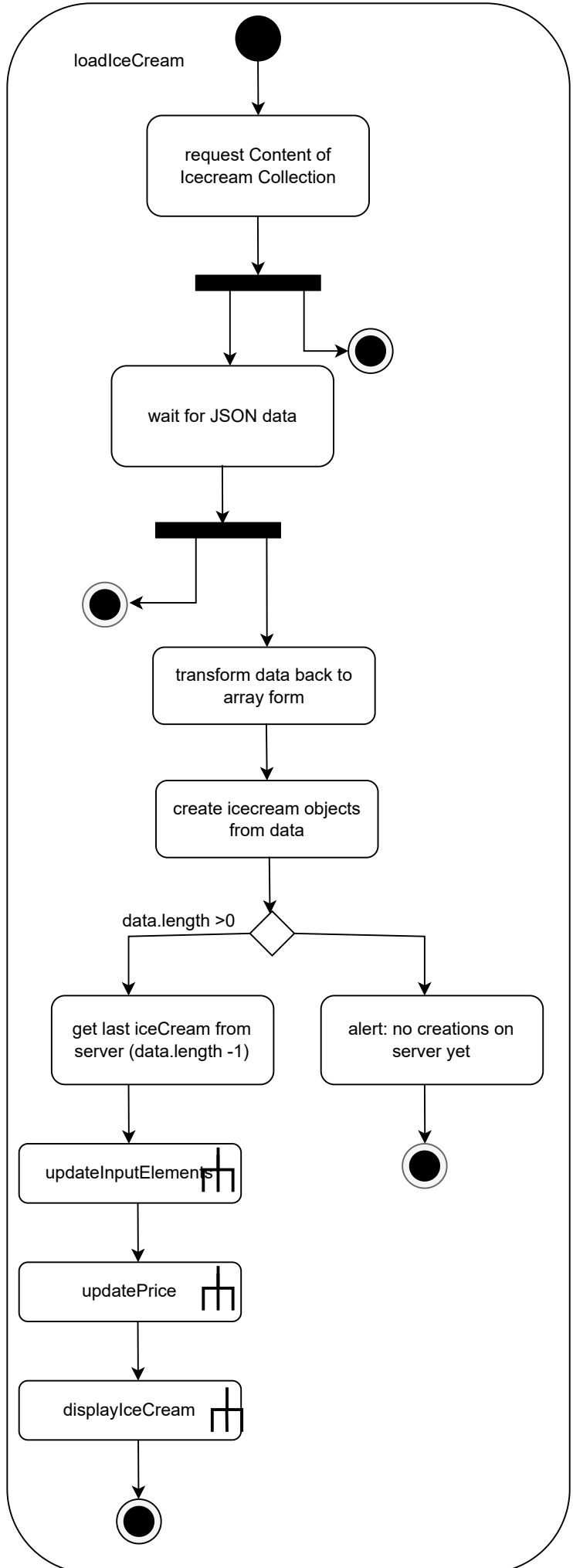




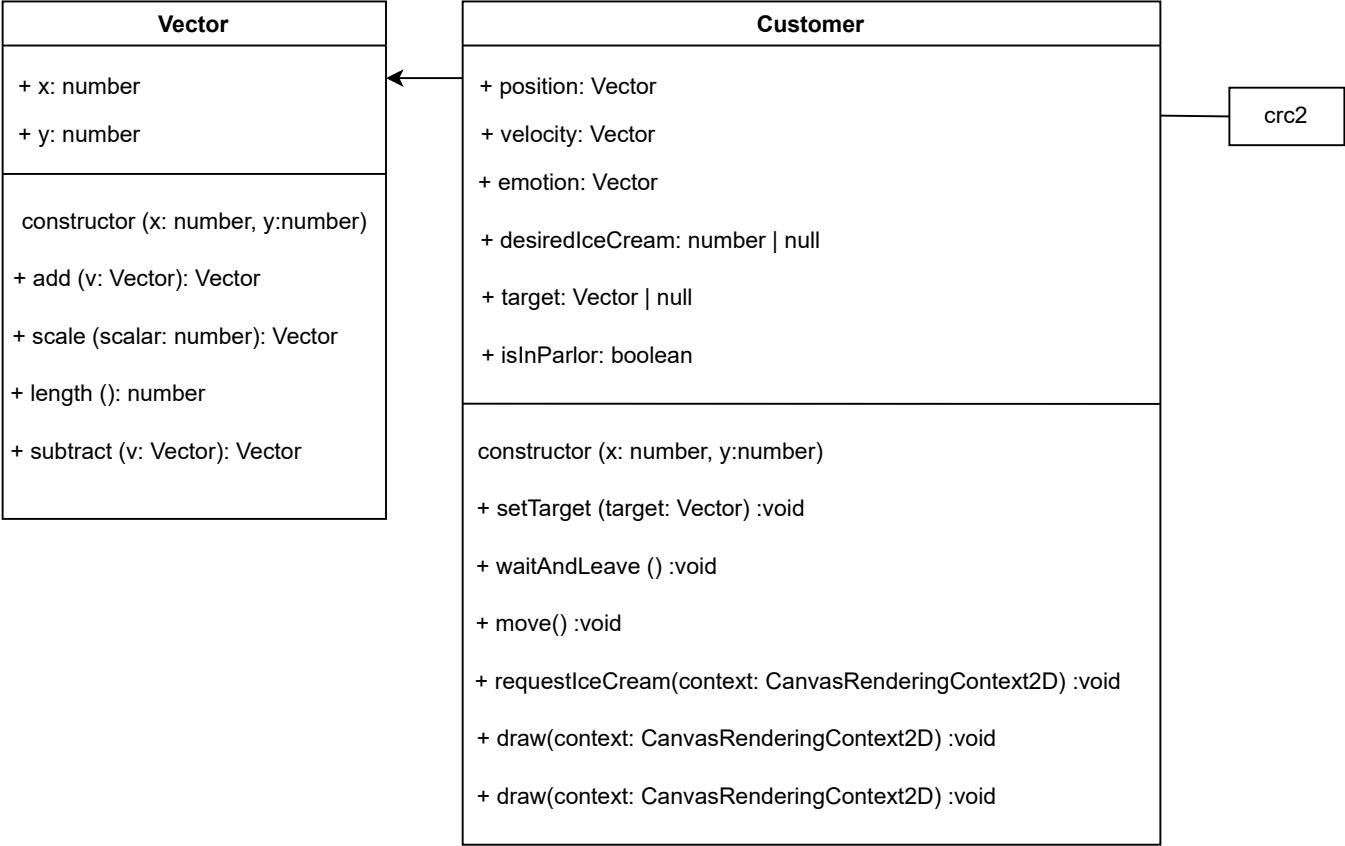
saveIceCream



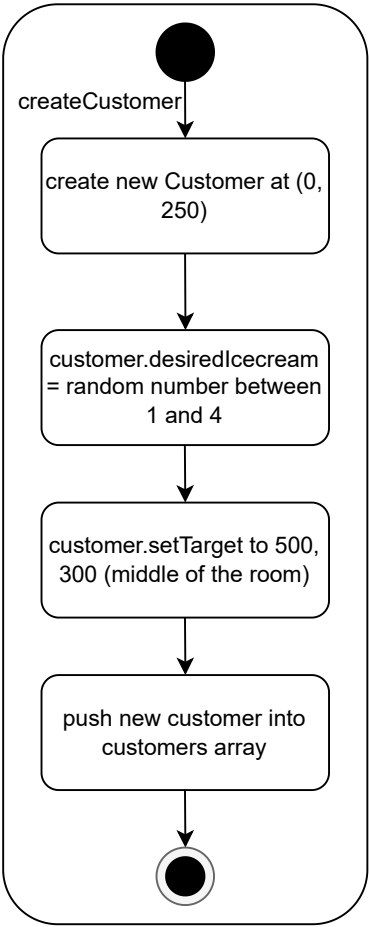
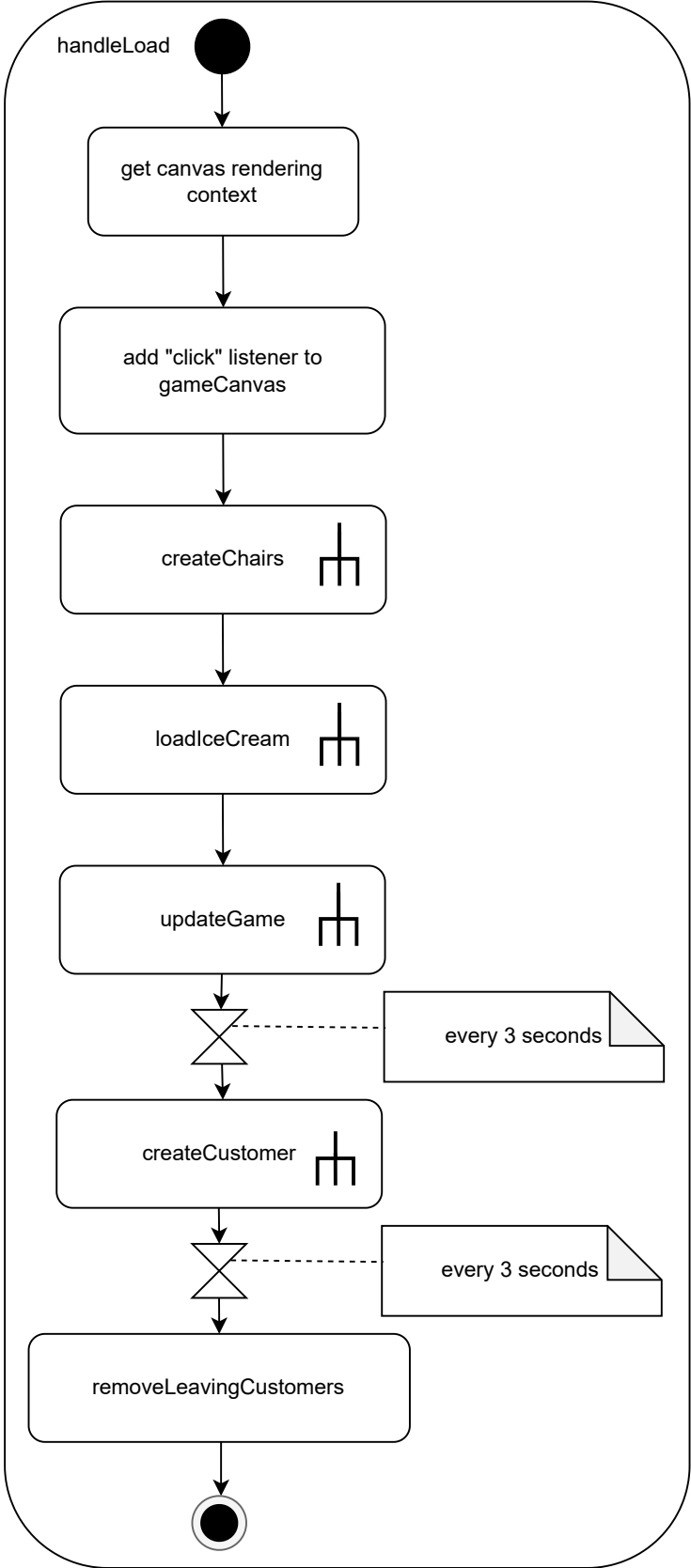
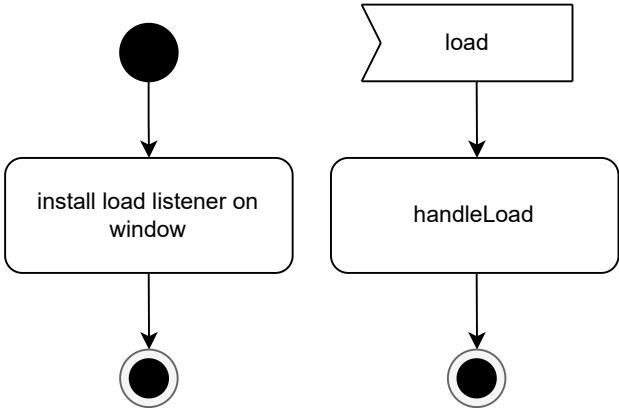
loadIceCream



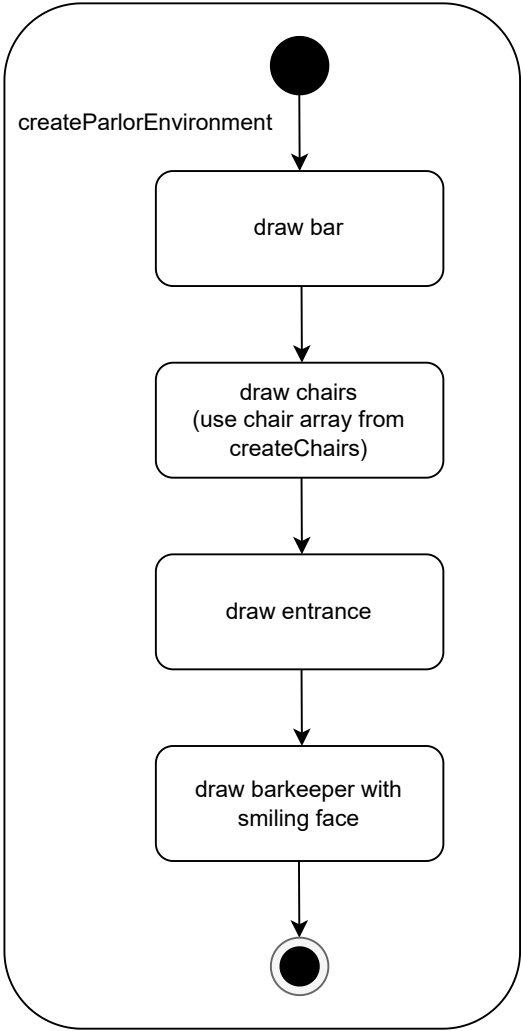
Eisdealer Classes

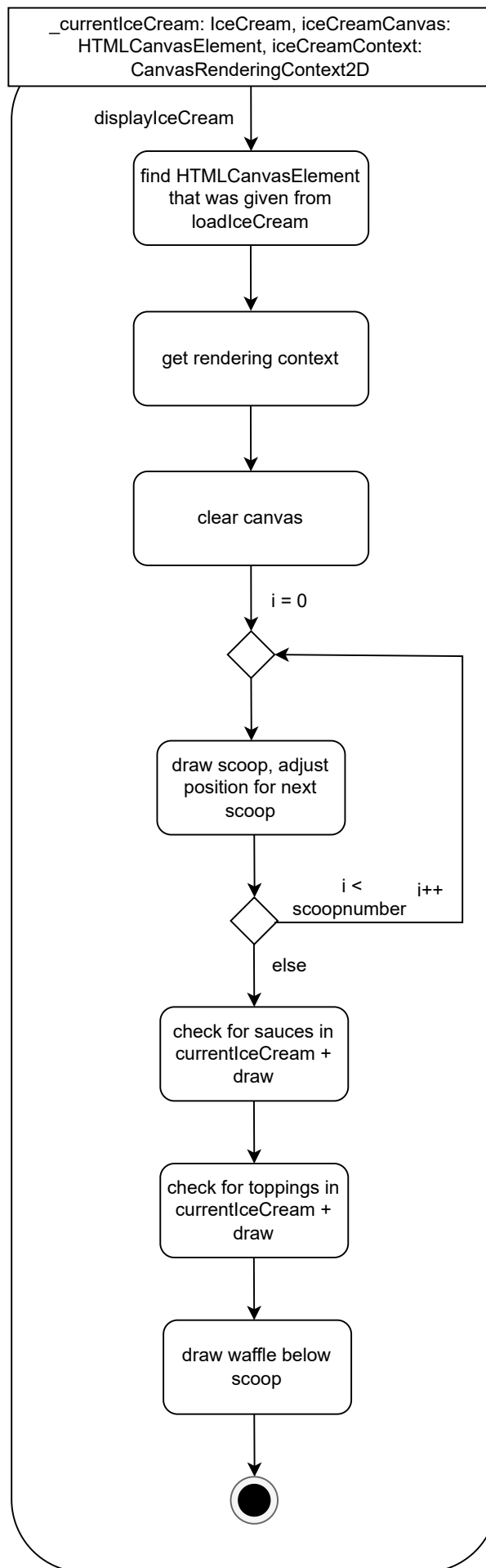
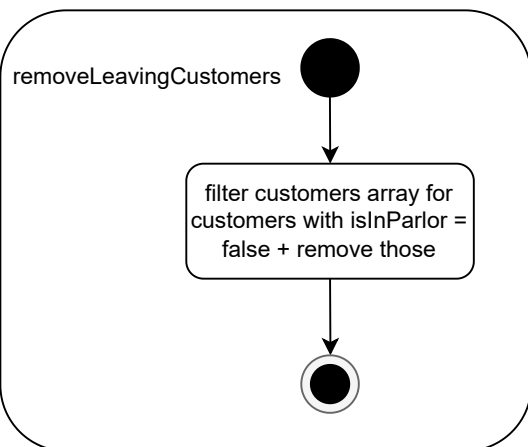
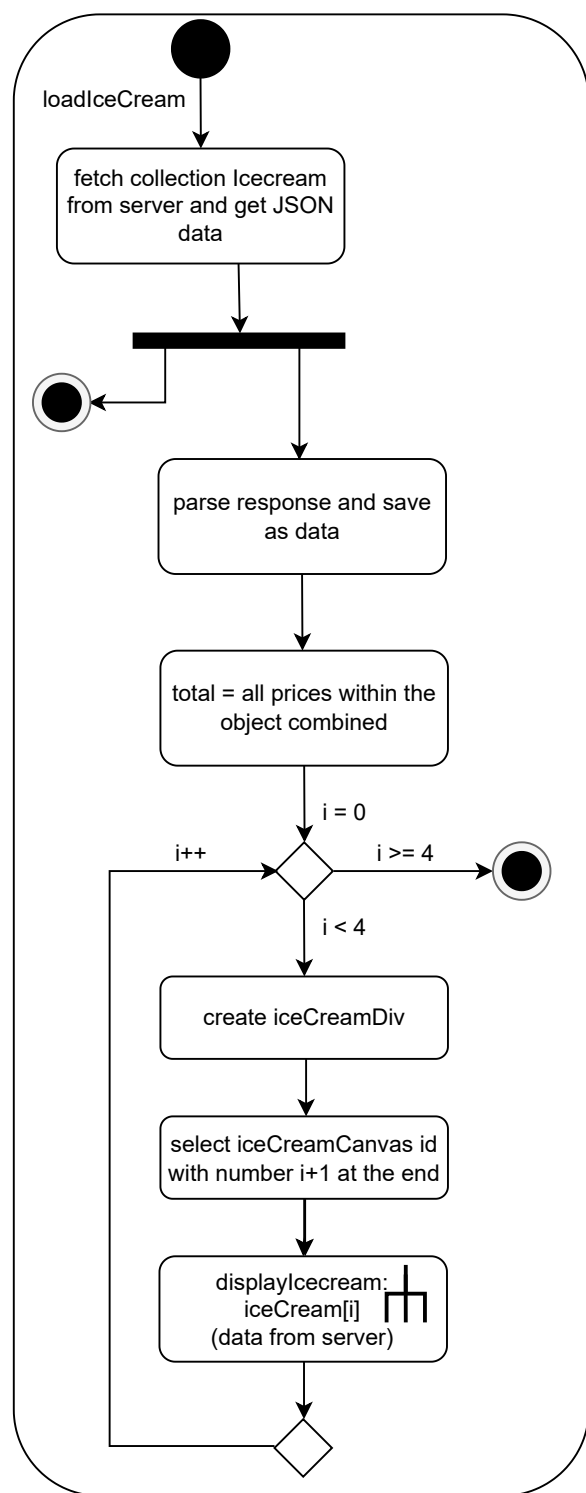


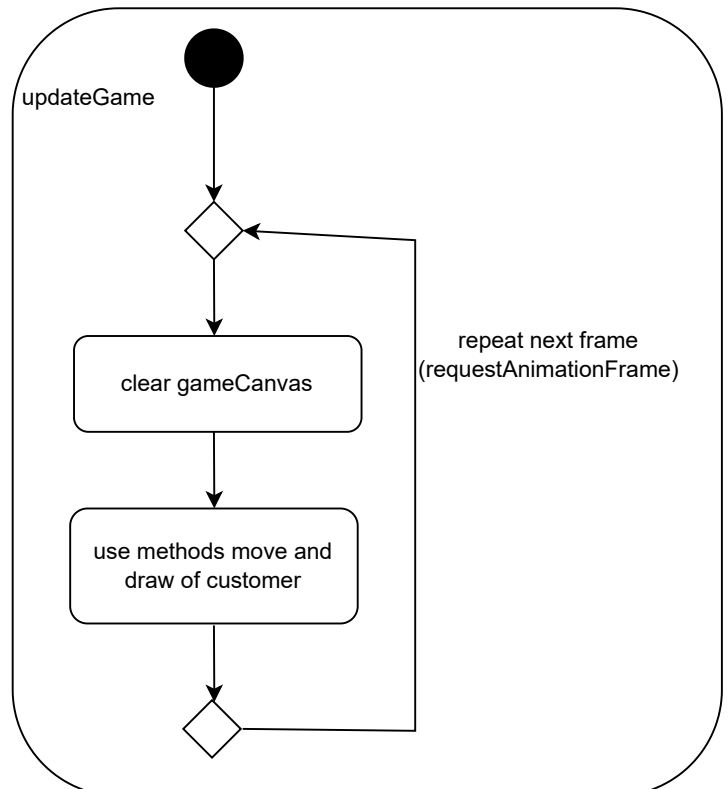
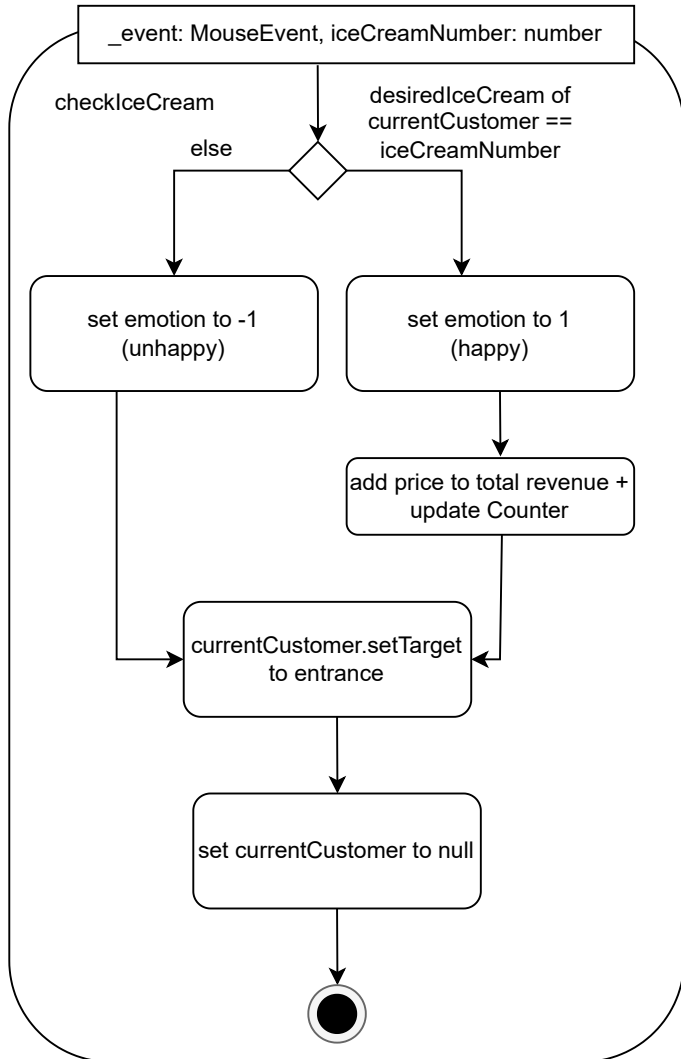
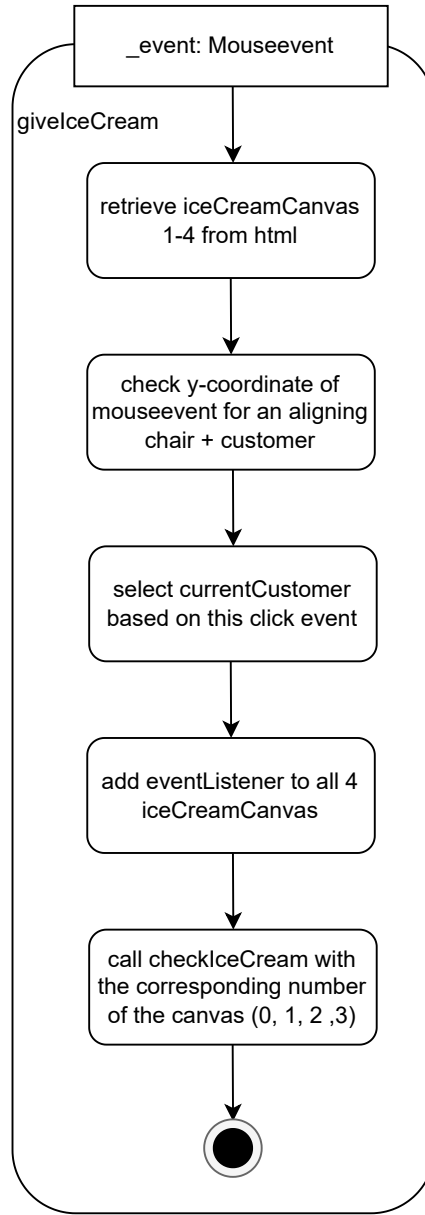
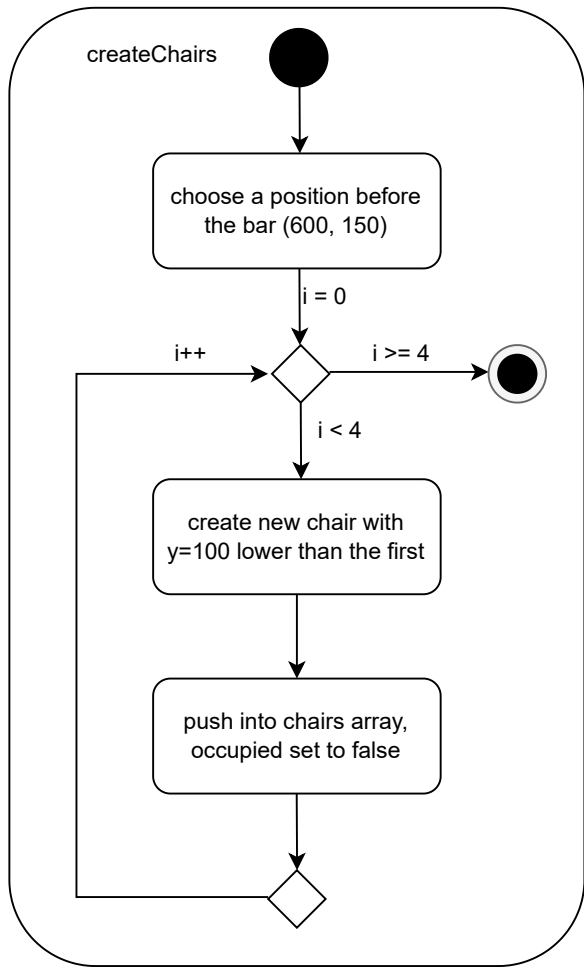
Eisdealer



Chair
position: Vector
occupied: boolean

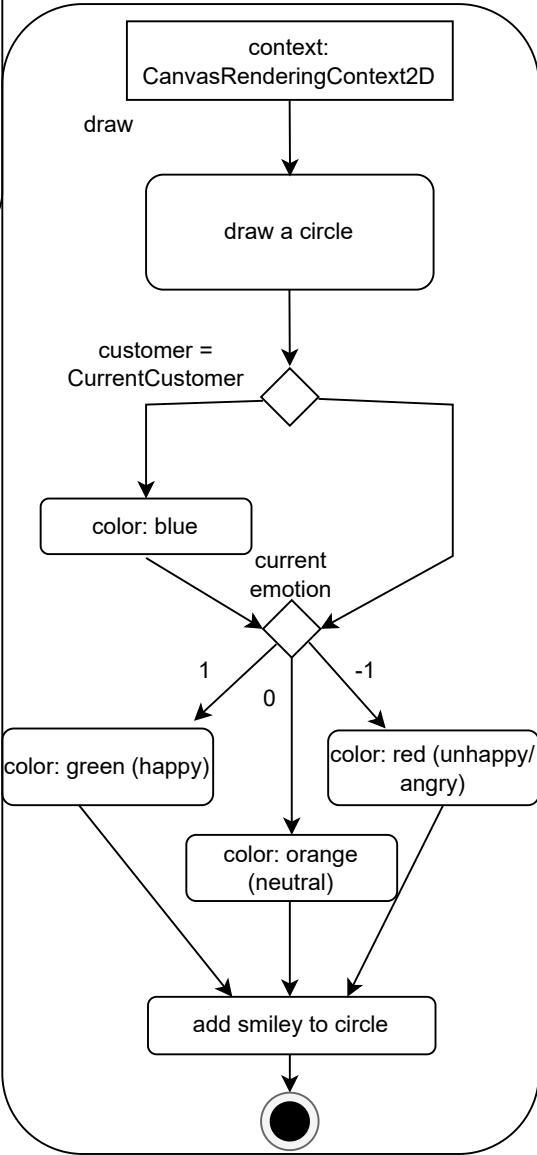
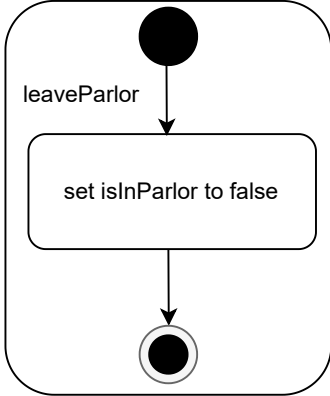
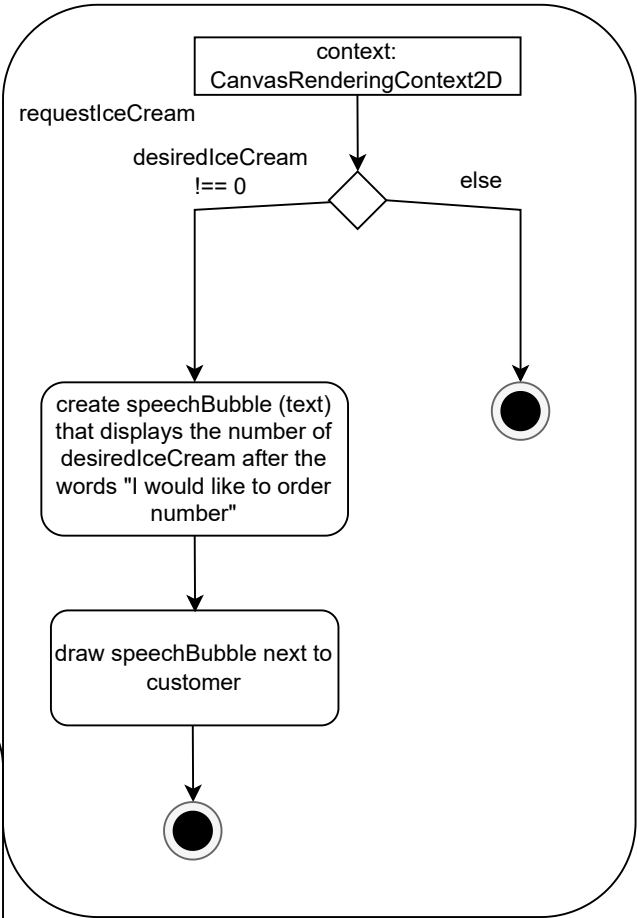
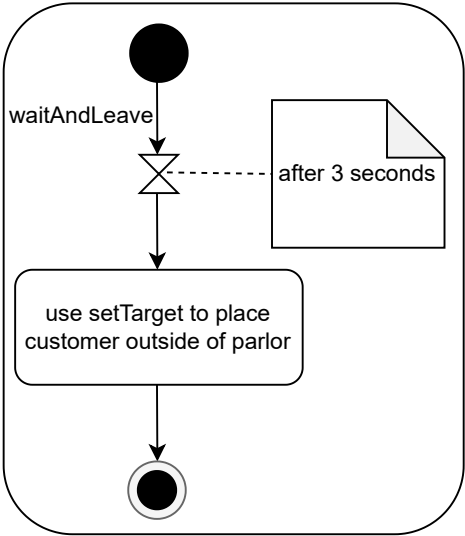
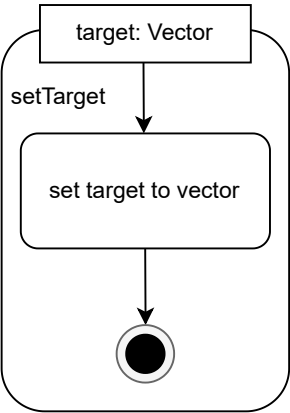


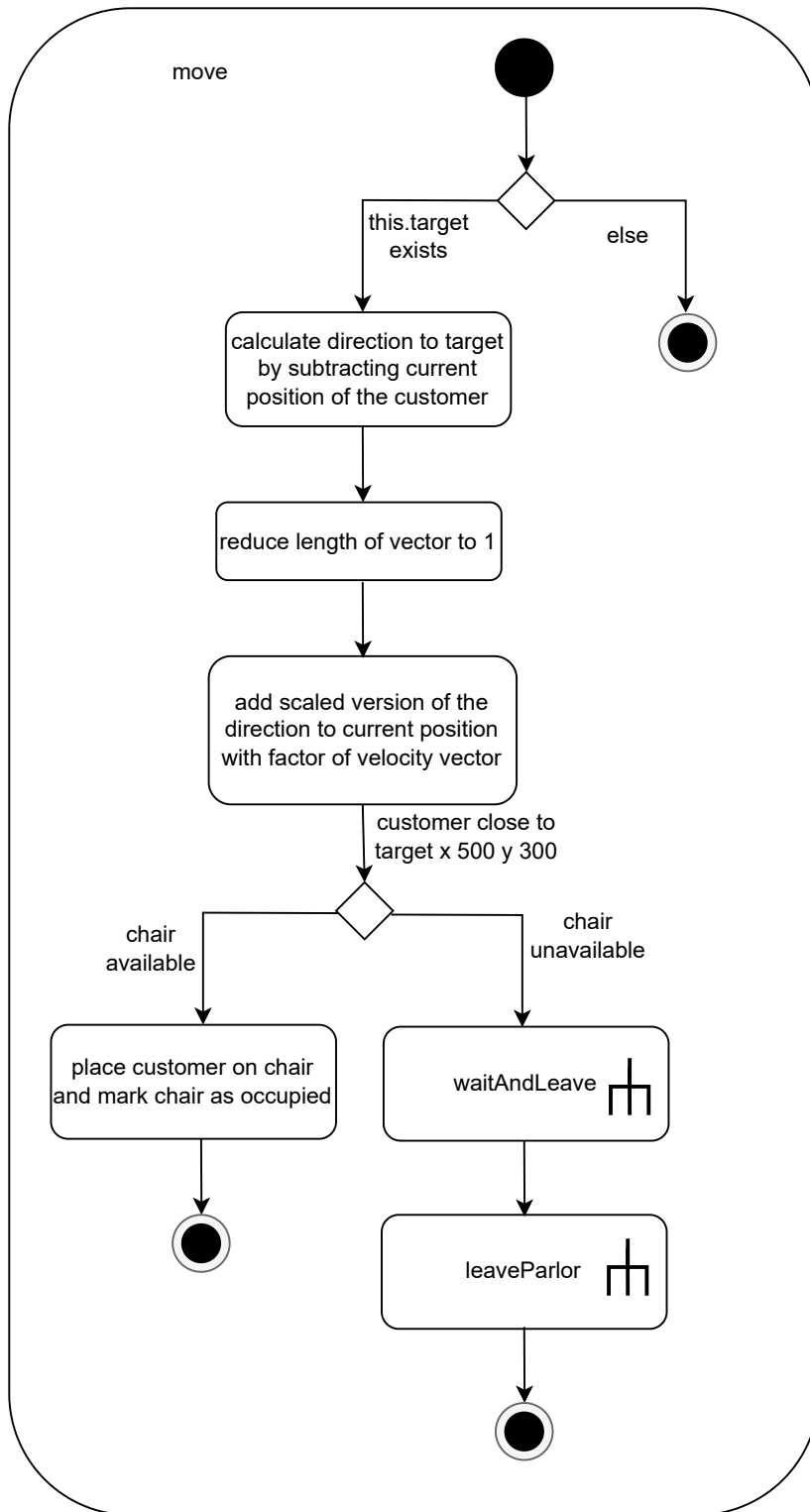




Eisdealer: Customer Class

Methods





Eisdealer: Vector Class

Methods

