

Zielplattform:

Das Spiel Eisdealer ist für die Anwendung an PCs konzipiert, als Maße für das Canvas wurde daher das Querformat mit 1920x1080 Pixel gewählt. Die Anwendung funktioniert sowohl per Maus als auch per Touch. Die Bildschirmgröße von Smartphones ist zu klein, um einen genauen Überblick über das Geschehen zu erhalten.

Nutzerinteraktion:

Zu Beginn des Spiels liest und versteht der Nutzer die Erklärung in der Alertbox. Nach Beginn des Spiels kann der Nutzer optional den Spiel-Soundtrack aktivieren. Zu Beginn kommen die Kunden noch sehr langsam in die Eisdiele, was dem Nutzer die Möglichkeit gibt, sich mit der Funktionsweise vertraut zu machen. Kommt ein Kunde in die Eisdiele, wird, sobald er an der Theke ankommt, eine Eisbestellung von einer bis drei Eiskugeln mit zufälligen Sorten angezeigt. Der Nutzer hat daraufhin die Aufgabe, die Bestellung in der richtigen Reihenfolge nachzubauen. Dafür klickt er auf die Eissorten auf der Theke, macht er dabei einen Fehler, kann er die Auswahl durch einen Klick auf das rote x wieder entfernen. Ist die Auswahl richtig, bestätigt er dies mit dem grünen Haken. Ist die Auswahl dennoch falsch wird das Spiel beendet und ein Alertfenster mit einer Erklärung und dem Endergebnis wird angezeigt. Ist die Auswahl richtig geht der Kunde mit seinem unsichtbaren Eis zu einem zufälligen Platz. Dort isst er das Eis und geht dann zur Kasse. Er kann jedoch schon zuvor vom Nutzer angeklickt werden, sodass er sich "beeilt". An der Kasse müssen die Kunden angeklickt werden. Eine Eiskugel kostet einen Euro, die Kunden bezahlen für die Anzahl der bestellten Eiskugeln und verlassen dann die Eisdiele. Es gibt jedoch eine Schwierigkeit, die Kunden sind sehr ungeduldig, sie wechseln nach einer gewissen Zeit zuerst in einen ungeduligen, orangenen und dann in einen wütenden roten Zustand. Beim grünen Zustand zahlen sie den vollen Geldbetrag, bei orange jedoch nur die Hälfte und bei rot gar nichts. Man sollte die Kunden also nicht zu lange warten lassen. Der Intervall mit dem die Kunden die Eisdiele Betreten nimmt mit laufender Spieldauer zu, es wird also immer schwieriger die Bestellungen zu erfüllen, zudem werden die alten Bestellungen immer von den neuen überschrieben, man kann also nicht sehen was der noch nicht abgearbeitete, vorherige Kunde wollte. Nach etwa zwei Minuten kommen so viele Kunden, dass man sie nicht mehr bedienen kann, dann ist es schlau, die essenden Kunden noch bezahlen zu lassen und durch eine falsche Eingabe das Spiel absichtlich zu beenden. Der am Ende angezeigte Geldbetrag bietet eine Vergleichbarkeit zu vorherigen Ergebnissen oder den Ergebnissen anderer an.

(1. Konzept)

EISDEALER



Kunden warten vor Eisdieler, wenn sie voll ist (alle 20 sek)

1 Kunden kommen zur Eisdieler und ^{Teke} ~~und setzen sich~~ und gehen zur Teke warten

2 Kunden wählen ^{direkt} Eissorten (1-3 mit unterschiedliche Reihenfolge)

Wird an der Kasse angezeigt

4 Nutzer stellt Bestellungen zusammen, Gast setzt sich und wartet

5 Nutzer gibt Bestellungen aus durch Haken oder Löscht durch Mülleimer
↓ Eis erscheint beim Kunden

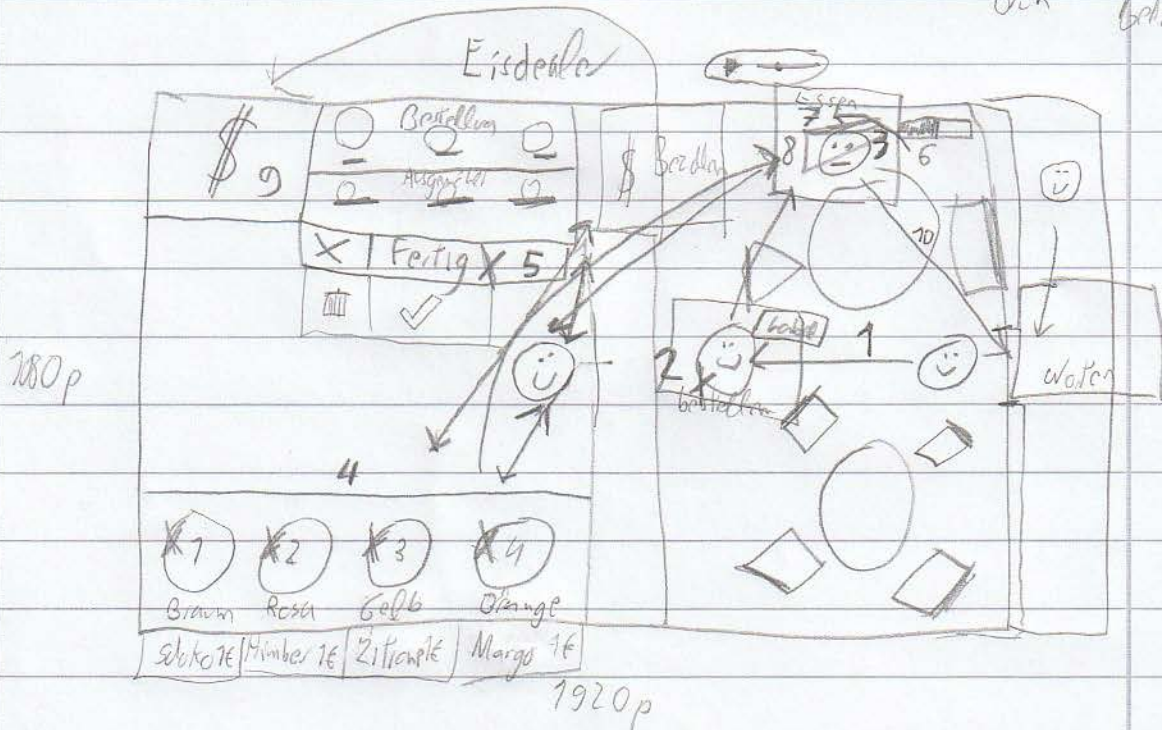
6 Kunden essen langsam Eis auf (20 sek / Balken?)

7 Kunden bestellen Rechnung ← Kunde geht zu Kasse ✓

8 Nutzer gibt Rechnung → 9 Geld steigt

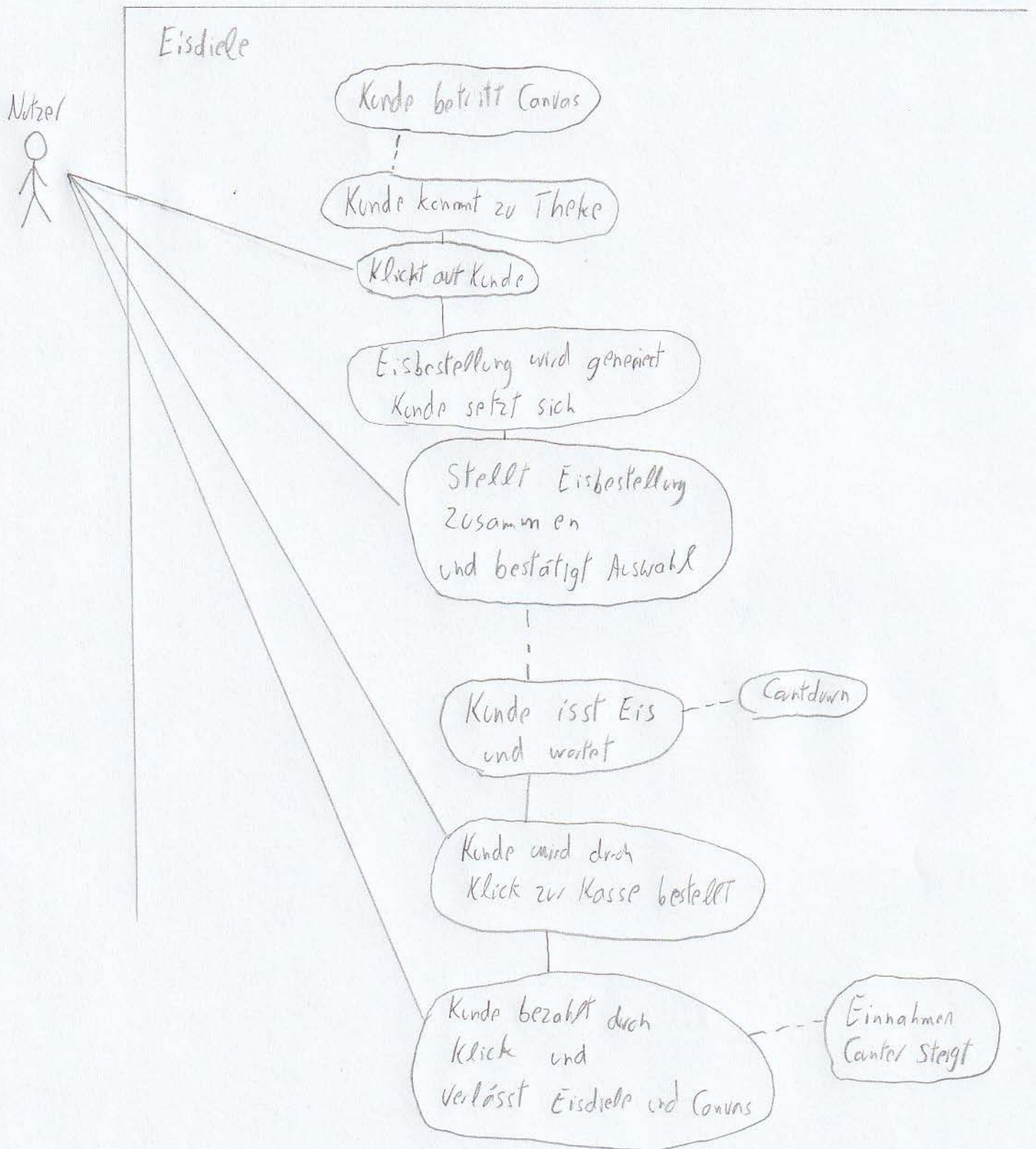
10 Kunden verschwinden

alle 20 sek
eine Emotion wählen



html / main / Movable / Drawables / Eisdieler / Kunden

USE - Case - Diagram



Konzept Dateistruktur - Eisdiel

(optional)
index.html
 - verbindet und sendet die js Dateien

style.css
 - Gestaltung.html

main.ts
 - steuert Spiel
 - zeichnet Spiel

middlewares.ts
 - move-Funktionen
 - bewegliche Objekte
 - Kunden

drawables.ts
 - Constructor
 - zeichnet alle festen Elemente

iceCream.ts
 "Eis" Klasse
 Array zur Speicherung
 - Schokolade
 - Himbeere
 - Zitrone
 - Mango
 + Farben und Preis

Customer.ts
 - Constructor
 - zufällige Bestellungs-Funktion
 - Stimmungen der Kunden

Vendor
 - Verkauf

Klassendiagramme

Drawable
draw(crc2): void

Movable
- position (x): number
- velocity (xy): number
- constructor: position velocity
- move: void
- abstract draw: crc2

Kunde: Attribute
- order: string
- waiting Time: number
- state: string
- smiley Color: string
- targets: xy: number
- mood Interval:
- has generated Order: boolean
- canvas Width/height: number

Zeichnungen
- draw (crc2): void
- draw Order (crc2): void
- draw selected Order: void
- draw Earnings: void

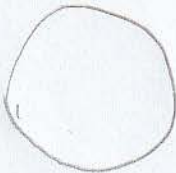

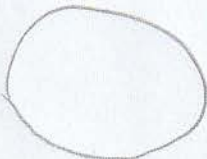



Aktualisierungen
- update Customers: void
- move Towards Target: void
- update Smiley: void

Bestellung
- generate Order: void
- get Order: string
- is Order Matching: boolean

Zeit
- increment Waiting Time: void
- reset Mood: void
- start Mood Timer: void
- stop Mood Timer: void

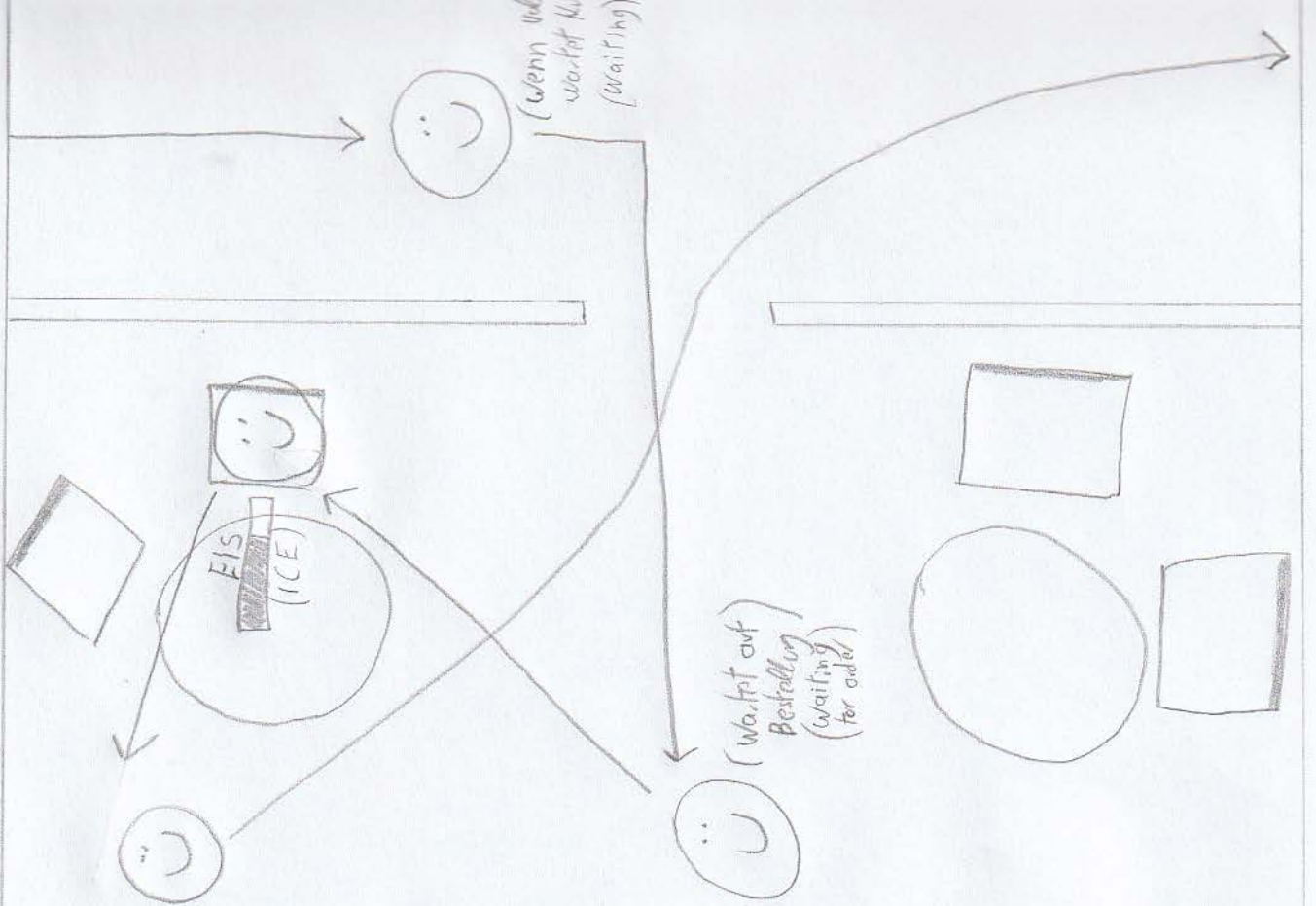
Zugriffe
- get State: string
- get Position: number
- find free Chair: number
- show Game Over Alert: void
- handle Payment: void

Canvas - Skizzo #1

Bestellung (Order)	
	
	
	
	x ✓

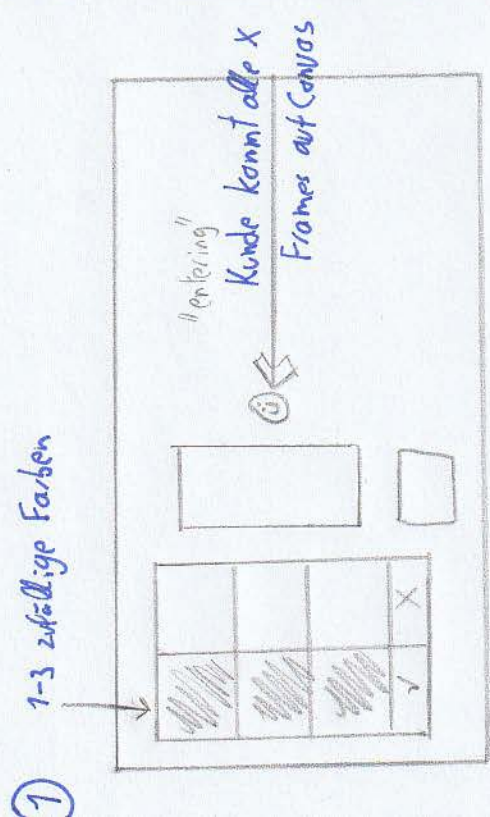


A hand-drawn diagram showing four colored squares arranged in a row. Each square is labeled with a fruit name above it and a color name inside it in parentheses. From left to right: a brown square labeled "Schokolade" with "(Brown)" inside; a pink square labeled "Himbeere" with "(Rosa)" inside; a yellow square labeled "Zitrone" with "(Gelb)" inside; and an orange square labeled "Mango" with "(Orange)" inside.

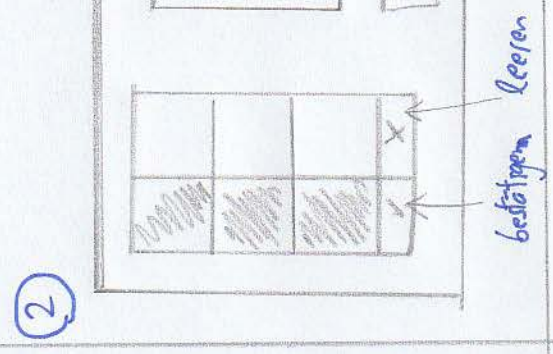


Skizze - Spielablauf

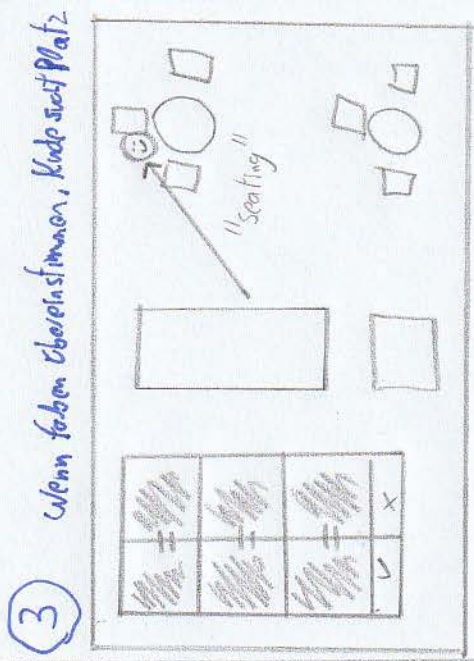
1-3 zufällige Farben



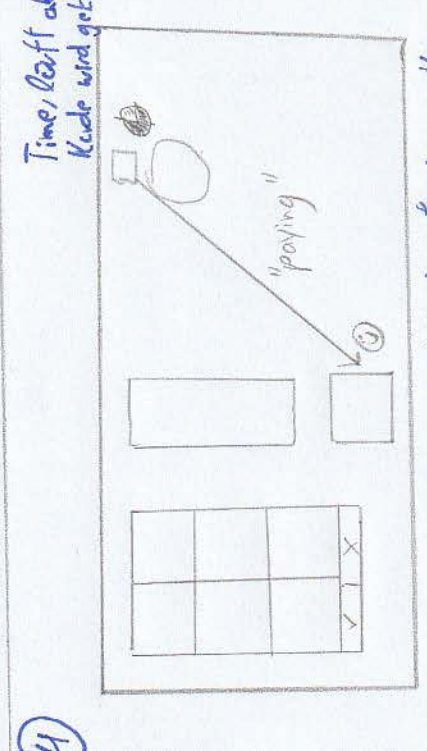
2 Auswahl der Farben auf Theke



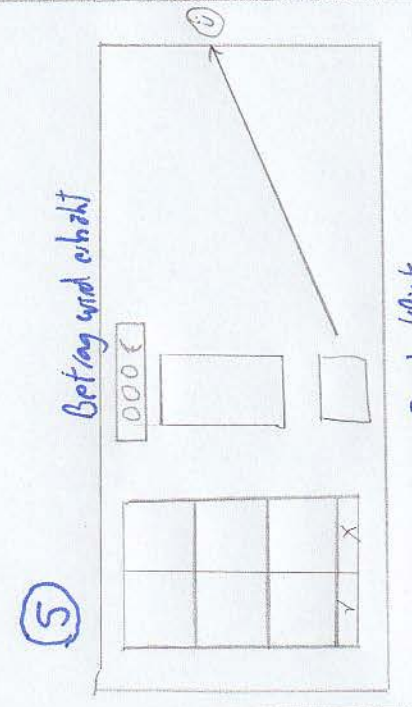
3

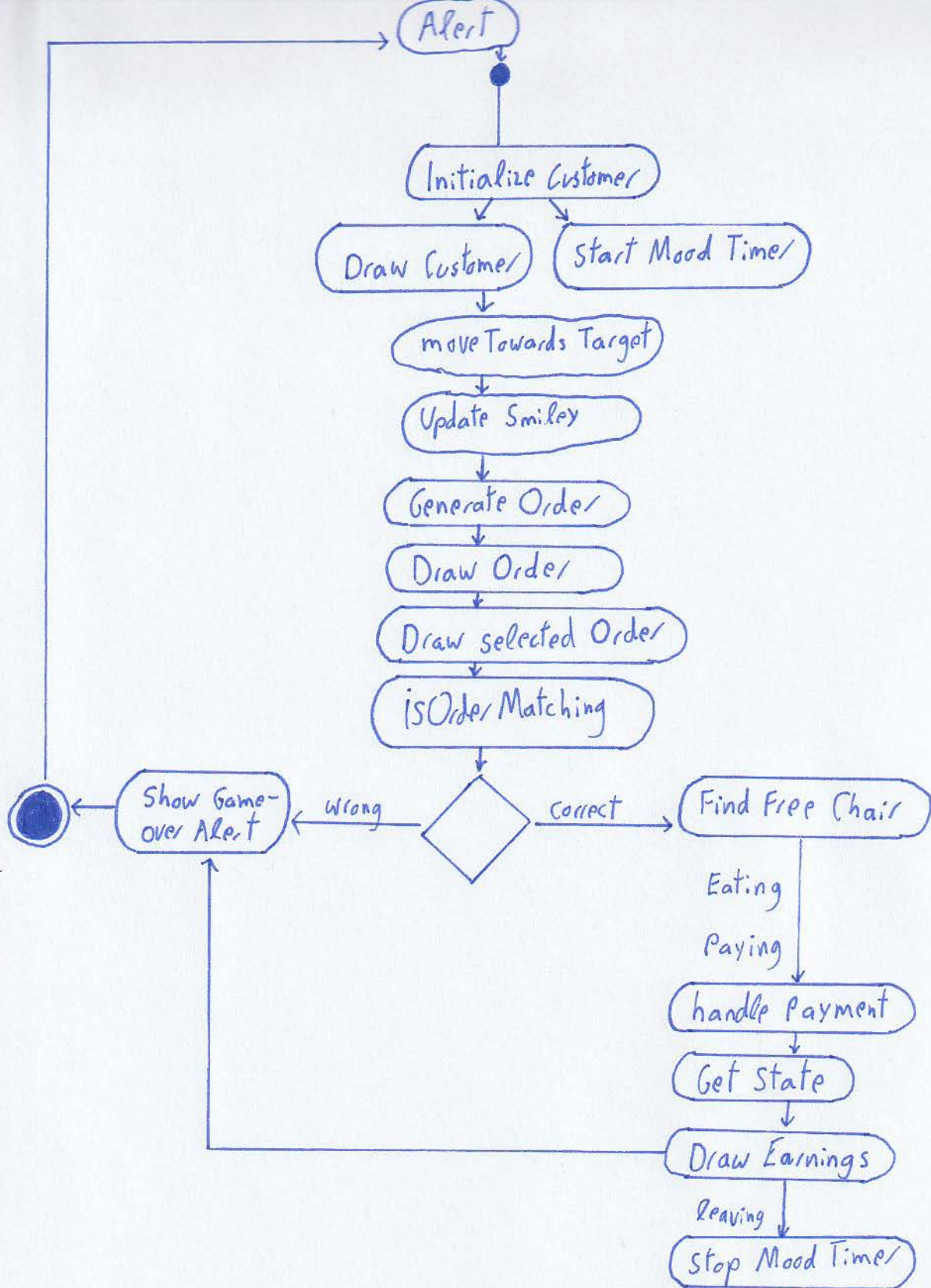


4 Time/Zeit ab oder Kunde wird geteakt

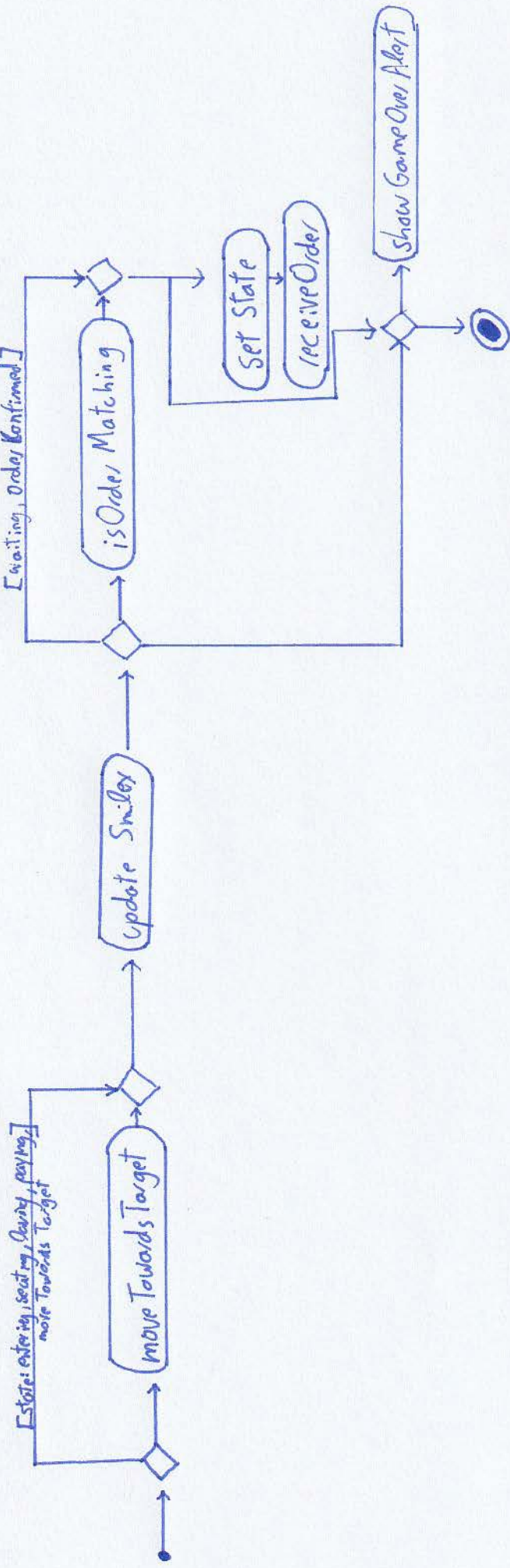


5 Betrag wird eingeht

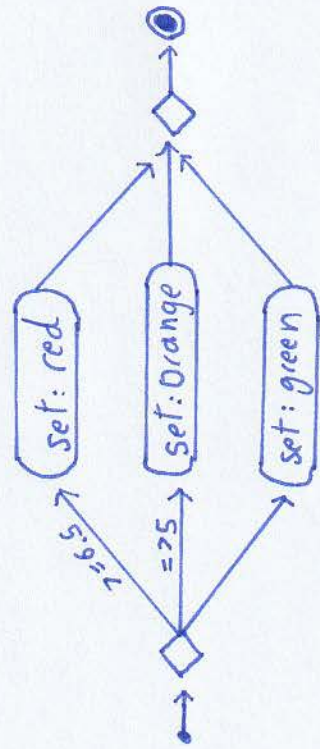




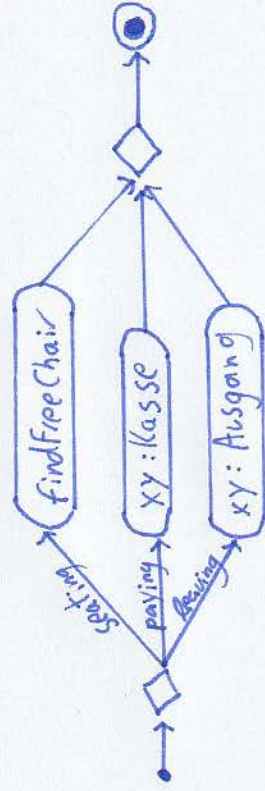
update



update Smiley (Stimmung)



set State



Testphase - Eisdiele

Name Testperson:	Problem/Reaktion:	Verbesserung:
Ole	-Skalierung auf Bildschirm stimmt nicht	Skalierung flexibel machen
	-Funktionalität wird nicht klar	Sprechblasen hinzufügen oder Audiodateien mit Erklärtext
	-Demotivation durch schnelles Scheitern	Einstiegslevel einführen für Erfolgserlebnis
	-gleiche Schwierigkeit ist langweilig	Erhöhen der Geschwindigkeit nach Dauer
	-Kritik an Github Ordnerstruktur	Ordner aufräumen
Sören	-Verwirrung durch Verzögerung bei generierten Eissorten	Erklärung hinzufügen
	-versteht Spielablauf	
Marius	-zu hektischer Spielverlauf	Erhöhen der Geschwindigkeit
Jannis	-versteht Spielablauf nicht	Erklärung hinzufügen
	-Vorschlag von Levelsystem	Level mit unterschiedlichen Geschwindigkeiten einbauen
	-Entdeckung von Bug, Kunden können auch ohne Eis zum Platz	Klickbarkeit in der Phase deaktivieren
	-Hitbox der Kunden ist zu klein	Auf größe des gesamten Smileys vergrößern
Alex (ich selbst)	-Text oben links im Canvas	im Code finden und entfernen
	-Eis Array im Drawables funktioniert nicht	beide Dateien verbinden (klappt noch nicht)
	-Preise ändern können	stelle im Code markieren
	-Soundtrack	Audiodatei einbauen