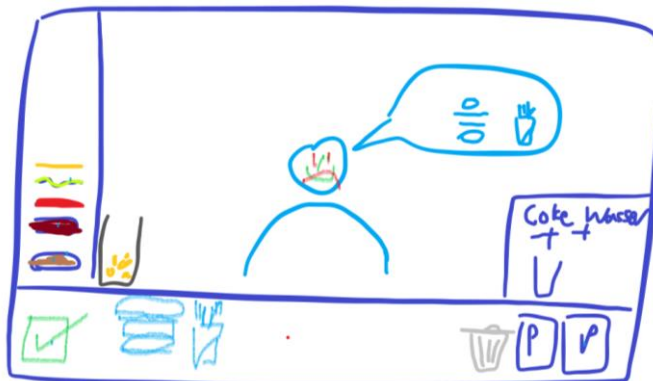


1 Teilnehmer/innen des Teams:

Klasse: BI19a	Team: BurgerPlace
------------------	----------------------

2 Anforderungsdefinition (Meilenstein A)

„BurgerPlace“	
Auftrag: (Allgemeine Beschreibung)	<p>Nutzen: Das Spiel dient zum reinen Zeitvertreib.</p> <p>Szenario:</p> <p>Das Spiel spiegelt den Arbeitsalltag eines Arbeiters in einem Burger-Restaurant wider. Der Spieler befindet sich als Koch in einem Restaurant. Es kommen Kunden mit Burgerbestellungen ins Restaurant. Der Spieler hat diverse Zutaten vor sich auf einer Arbeitsfläche und kann diese zum bestellten Burger kombinieren. Diesen übergibt er dem Kunden und bekommt einen Geldbetrag dafür. Je schneller der Spieler den Burger zusammenstellt, desto mehr Geld bekommt er. Kombiniert der Spieler falsche Zutaten oder stellt sie in der falschen Reihenfolge zusammen, bekommt er keine Bezahlung.</p> <p>Details:</p> <ul style="list-style-type: none"> • Personen bestellen einen Burger • Der Burger muss korrekt zusammengestellt und der Person übergeben werden • Spieler bekommt Geld (Punkte) für das schnelle und korrekte Zusammenstellen des Burgers. <p>Machbarkeitsabklärung:</p> <ul style="list-style-type: none"> • Mockup erstellt • Es existieren bereits viele Versionen solcher Spiele, daher sollte eine Umsetzung auch in Greenfoot möglich sein



MUSS Kriterien: (Konkrete Features, die umzusetzen sind)	Folgende Features sollen implementiert werden (Funktionalität): <ul style="list-style-type: none">• Kunden kommen in den Laden und bestellen einen Burger.• Im Interface befinden sich die Zutaten, die man in der richtigen Reihenfolge auswählen muss, um den gewünschten Burger zu bauen.• Wenn der Burger korrekt serviert wurde, bezahlt der Kunde diesen. Je länger man zum Bauen des Burgers braucht, desto weniger Trinkgeld bekommt man.• Das Spiel endet nach einer bestimmten Zeit.
KANN Kriterien: (Konkrete Features, die optional sind)	Folgende Features können zusätzlich implementiert werden: (Kreativität) <ul style="list-style-type: none">• Die Zutaten kosten Geld, man darf nicht Konkurs gehen• Der Spieler hat die Möglichkeit mehrere Arbeitstage hintereinander zu spielen und behält dazwischen das erarbeitete Geld• Option auch Getränke und Beilagen wie Pommes oder Onionrings zu bestellen• Zutaten können per "drag and drop" bewegt und zusammengestellt werden.

2.1 Planung LB2

<i>MS</i>	<i>Tätigkeit / Abgabe</i>	<i>Soll-Datum</i>	<i>Ist-Datum</i>
A	Projektstart <ul style="list-style-type: none">➤ Team Bildung➤ Wahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson		
B	Teamaufgabe 1: <ul style="list-style-type: none">➤ Abgabe: Lösungsdesign (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard)		
B2	Teamaufgabe 2: <ul style="list-style-type: none">➤ Abgabe: Testvorschrift und Testfälle		
C	Einzelaufgabe 3: <ul style="list-style-type: none">➤ Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)➤ Fachgespräch Projektabschluss		
C2	Einzelaufgabe 4: <ul style="list-style-type: none">➤ Abgabe: Ausgefüllter Systemtest		

3 Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

3.1 Funktionsmodell

Im Folgenden sind die erwarteten Eingaben und Ausgaben beschrieben / dargestellt:

...

(Funktionsmodell: Skizze, Bild, Mindmap, Blockdiagramm zur obigen Anforderungsdefinition mit Legende)

Legende:

-

3.2 Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:

...

(Detaillierte Beschreibung der UseCases)

Legende:

...

3.3 Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

...

(Storyboard)

4 Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument

[M226B_LB2_Testvorschrift_MS-B2.docx](#)

5 Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Das erstellte Java-Projekt (Greenfoor-Szenario) ist hier detailliert abgelegt:

[M226B_Aufgabe_3_Szenario_IhrName.zip](#)

5.1 Statisches Design: Klassendiagramm

Folgend die statische Struktur des Szenarios

...

(UML Klassendiagramm mit Assoziationen und Kardinalitäten)

5.2 Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

...

(Umstände / Anpassungen / Veränderungen)

5.3 Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen detailliert beschrieben:

...

(Ausführliche Beschreibung der internen Funktionen
oder Verweis zum Inline-Kommentar mit JavaDoc! (`/** @param @return */`))

5.4 Dynamische Struktur: Sequenzdiagramm

Ein zentraler Ablauf eines UseCases ist im Folgenden dargestellt:

...

(Darstellung eines zentralen Ablaufs mittels Sequenzdiagramm)

Trace: ...

...

6 Bedienungsanleitung (Meilenstein C: individuelle Aufgabe 3)

...

7 Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument

[**M226B_LB2_Testvorschrift_MS-C2_Name.docx**](#)