

Systemtest anhand der Anwendungsfallszenarien**1 Teilnehmer/in des Teams:**

| | |
|-----------------------------|------------------------------|
| Name: Fankhauser Lenz | Vorname: Daniel Sandro |
|-----------------------------|------------------------------|

| | | |
|---------------|------------------|----------------------|
| Abgabedatum : | Klasse: BI19a | Team: BurgerPlace |
|---------------|------------------|----------------------|

2 Testbeschreibung**2.1 Ziel des Tests**

Testen des Spiels «BurgerPlace» anhand der Use Cases

2.2 Art des Tests

Blackbox-Test

2.3 Verwendete Hilfsmittel**2.4 Anforderung an das Testobjekt**

Greenfoot installiert und funktional

2.5 Testvorgaben**2.6 Abbruchkriterien**

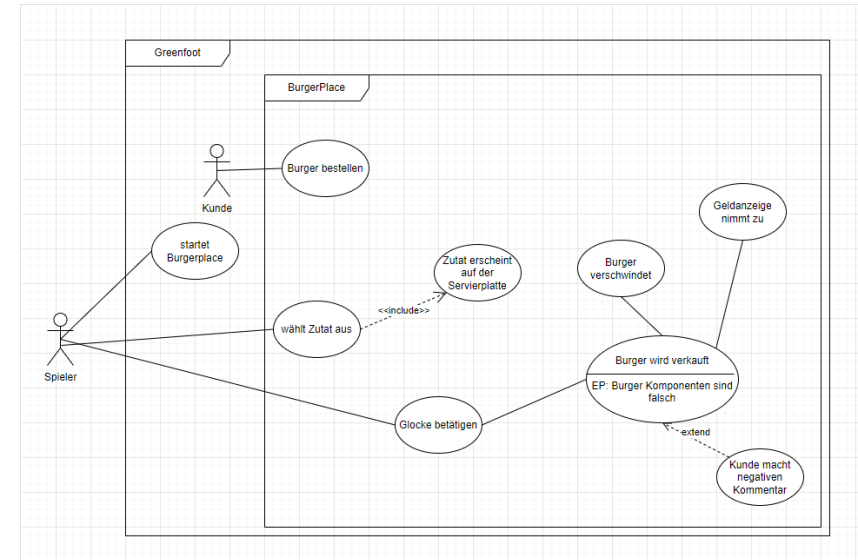
Gravierender Fehler im Programm

2.7 Weiteres

Systemtest anhand der Anwendungsfallszenarien

3 Testprotokoll - Testvalidierung

| | |
|---|---|
| Projektname | <i>BurgerPlace</i> |
| Version (getestetes Programm) | <i>v1.0</i> |
| Projekt-Code (Dateien) | https://github.com/sandrolenz/M226b-BurgerPlace |
| Fachlicher Ansprechpartner (Namen der Lehrperson) | <i>Kellenberger Michael</i> |
| Autor des Testprotokolls | <i>Fankhauser Daniel, Lenz Sandro</i> |
| Testdatum | |
| Name Tester | |



| Use-Case | | Testfall | | | |
|---|---------------------------|---|--|------------------------|----|
| UC „Burger bestellen“: | | Test-Case “Burger bestellen“: | | | |
| Akteure: Kunde, GF-Engine Precondition: Restaurant ist leer Ereignis: Neue Bestellung wird erstellt | | Trace 01: Korrekter Durchlauf ohne Fehler | | | |
| # | Ablauf UC | Testaktivität (Input) | Erw. Resultat System/Benutzer | Tatsächliches Resultat | OK |
| 1 | Laden leer | Kunde erscheint | Kunde kommt durch die Tür | | |
| 2 | Bestellung wird angezeigt | Bestellung generieren und zwischen-speichern | Sprechblase mit Bestellung wird an-gezeigt | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| Postcondition: Spieler beginnt mit Herstellung des Burgers | | Postcondition: Spieler kann Burger herstellen | | | |

Systemtest anhand der Anwendungsfallszenarien

| Use-Case | | Testfall | | | |
|--|------------------------------|--|--|------------------------|----|
| UC „Zutaten auswählen“: | | Test-Case “Burger zubereiten“: | | | |
| Akteure: Spieler Precondition: Kunde ist im Restaurant und hat Bestellung aufgegeben Ereignis: Spieler bereitet Burger zu | | Trace 01: Korrekter Durchlauf ohne Probleme | | | |
| # | Ablauf UC | Testaktivität (Input) | Erw. Resultat System/Benutzer | Tatsächliches Resultat | OK |
| 1 | Teller leer, Zutat auswählen | Spieler wählt erste Zutat aus und platziert sie per drag&drop auf dem Teller | Erste Zutat wird auf den Teller gelegt | | |
| 2 | Weitere Zutaten auswählen | Spieler wählt weitere Zutaten aus und platziert sie auf dem Teller | Zutaten werden zum Burger hinzugefügt. | | |
| 3 | Burger ist fertig | Spieler betätigt die Klingel wenn der Burger fertig ist. | Klingel spielt einen Ton ab | | |
| Postcondition: Burger kann mit Bestellung überprüft werden | | Postcondition: Fertiger Burger ist zur Abgabe bereit | | | |

| Use-Case | | Testfall | | | |
|--|------------------------------|--|--|------------------------|----|
| UC „Zutaten auswählen“: | | Test-Case “Burger zubereiten“: | | | |
| Akteure: Spieler Precondition: Kunde ist im Restaurant und hat Bestellung aufgegeben Ereignis: Spieler bereitet Burger zu | | Trace 02: aus Versehen falsche Auswahl einer Zutat | | | |
| # | Ablauf UC | Testaktivität (Input) | Erw. Resultat System/Benutzer | Tatsächliches Resultat | OK |
| 1 | Teller leer, Zutat auswählen | Spieler wählt erste Zutat aus und platziert sie per drag&drop auf dem Teller | Erste Zutat wird auf den Teller gelegt | | |
| 2 | Weitere Zutaten auswählen | Spieler wählt eine falsche Zutat aus und platziert sie auf dem Teller | Zutat wird zum Burger hinzugefügt. | | |

Systemtest anhand der Anwendungsfallszenarien

| | | | | | |
|---|-------------------|---|---|--|--|
| 3 | Teller leeren | Spieler klickt auf den Abfall-Button | Der Burger verschwindet, es wird Geld für die Zutaten abgezogen | | |
| 4 | Zutat auswählen | Spieler wählt korrekte Zutaten aus und platziert sie auf dem Teller | Zutaten werden zum Burger hinzugefügt. | | |
| 5 | Burger ist fertig | Spieler betätigt die Glocke, wenn der Burger fertig ist. | Glocke spielt einen Ton ab | | |
| Postcondition: Burger kann mit Bestellung überprüft werden | | Postcondition: Fertiger Burger ist zur Abgabe bereit | | | |

| Use-Case | | Testfall | | | |
|---|-----------------------|---|---|------------------------|----|
| UC „Glocke betätigen“: | | Test-Case „Zubereiteter Burger ist fertig“: | | | |
| Akteure: Spieler, GF Precondition: Burger wurde zubereitet Ereignis: Burger wird überprüft | | Trace 01: Burger stimmt mit Bestellung überein | | | |
| # | Ablauf UC | Testaktivität (Input) | Erw. Resultat System/Benutzer | Tatsächliches Resultat | OK |
| 1 | Burger ist fertig | Spieler betätigt Glocke | Glocke spielt einen Ton ab | | |
| 2 | Burger wird überprüft | Greenfoot vergleicht den Burger mit der Bestellung | Burger und Bestellung stimmen überein, in der Sprechblase wird ein Haken angezeigt. | | |
| 3 | Burger stimmt überein | Kunde bezahlt den Burger und gibt ggf. Trinkgeld | Der Kontostand erhöht sich | | |
| 4 | Burger wurde bezahlt | Kunde verlässt den Laden | Kunde und Burger verschwinden, Sprechblase wird gelöscht | | |
| Postcondition: Der Laden ist leer | | Postcondition: Ein neuer Kunde kann den Laden betreten | | | |

Systemtest anhand der Anwendungsfallszenarien

| Use-Case | | Testfall | | | |
|---|-----------------------------|--|---|------------------------|----|
| UC „Glocke betätigen“: | | Test-Case “Zubereiteter Burger ist fertig “: | | | |
| Akteure: Precondition: Burger wurde zubereitet Ereignis: Burger wird überprüft | | Trace 02: Burger stimmt nicht mit Bestellung überein | | | |
| # | Ablauf UC | Testaktivität (Input) | Erw. Resultat System/Benutzer | Tatsächliches Resultat | OK |
| 1 | Burger ist fertig | Spieler betätigt Glocke | Glocke spielt einen Ton ab | | |
| 2 | Burger wird überprüft | Greenfoot vergleicht den Burger mit der Bestellung | Burger und Bestellung stimmen nicht überein, in der Sprechblase wird ein Kreuz angezeigt. | | |
| 3 | Burger stimmt nicht überein | Spieler klickt auf den Abfall-Button | Der Burger verschwindet, es wird Geld für die Zutaten abgezogen | | |
| Postcondition: Teller ist leer, der Kunde merkt sich den Fehler und wird kein Trinkgeld geben | | Postcondition: Spieler kann neuen Burger herstellen | | | |

LB2 Meilenstein B2 Teamaufgabe 2 / Meilenstein C2 Einzelaufgabe 4

Review des Testbeschriebs durch den Tester:

*(Tester beurteilt Testbeschreibung nach erfolgter Ausführung.
Fehler in der Beschreibung?
Fehler im Protokoll?)*

4 Sign-Off**Mängelliste:**

(Alle nicht mit OK markierten Testfälle hier auflisten und etwaige Beobachtungen und/oder Bemerkungen notieren, damit der Entwickler Anhaltspunkte zur Verbesserung erhält.)

- Test-Case _ Trace _:

Der Test

☐ wird **erfolgreich** abgenommen.

☐ wird eingeschränkt abgenommen (Mängel siehe oben).
Der Test wird **trotzdem als erfolgreich** abgenommen erklärt.

☐ wird **nicht** abgenommen (aufgetretene Mängel siehe oben)

Bis zum angegebenen Zeitpunkt werden alle oben beschriebenen Mängel beseitigt.

☐ Datum:

Test ist beendet und wurde korrekt durchgeführt

Ja ☐

Nein ☐

Unterschrift (Datum, Name *Tester*)

Ja ☐

Nein ☐

Unterschrift (Datum, Name *Autor*)

Validierung

Ja ☐

Nein ☐

Unterschrift (Datum, Name *Experte*)