



Mastermind

Testplan

**Hochschule Luzern
Technik & Architektur**

Programmieren 2 – FS12

Gruppe 10

Redzepe Iljasa | Reichmuth Marco | Rey Philipp | Rohrer Felix

Eine interdisziplinäre Projektarbeit
der Studiengänge Elektrotechnik und Informatik.

Horw, 11.05.2012

Autoren

Redzepi Iljasa	
Studiengang	Informatiker (Vollzeit)
Adresse	
Telefon	
E-Mail	iljasa.redzepi@stud.hslu.ch

Reichmuth Marco	
Studiengang	Elektroniker (Berufsbegleitend)
Adresse	
Telefon	
E-Mail	marco.reichmuth@stud.hslu.ch

Rey Philipp	
Studiengang	Elektroniker (Vollzeit)
Adresse	
Telefon	
E-Mail	philipp.rey@stud.hslu.ch

Rohrer Felix	
Studiengang	Informatiker (Berufsbegleitend)
Adresse	
Telefon	
E-Mail	felix.rohrer@stud.hslu.ch

Änderungskontrolle

Version	Datum	Autor	Beschreibung
1.0	10.05.2012	Marco Reichmuth	1. Version
1.1	11.05.2012	Felix Rohrer	JUnit Test Klassen ergänzt
-	11.05.2012	Felix Rohrer	Version 1.1 freigegeben

Inhalt

1	Einführung.....	1
1.1	Kontext	1
1.2	Testphilosophie	1
2	Begriffe, Abkürzungen und Referenzen	1
2.1	Begriffe & Abkürzungen	1
2.2	Referenzen	1
	Testplanung.....	2
2.3	Unit-Tests	2
2.4	System-Tests.....	2
3	Test-Design.....	2
4	Testfälle.....	3
	Abbildungsverzeichnis.....	6

1 Einführung

1.1 Kontext

Ziel ist es ein Programm zu erstellen, bei welchem sich der Kunde nicht um Fehler resp. Unregelmässigkeiten herumschlagen muss. Daher ist es wichtig die Software zu testen, um mögliche Fehler bereits frühzeitig zu erkennen.

1.2 Testphilosophie

Die Testphilosophie besteht aus den folgenden Punkten:

- Testen der Komponenten (Unit-Tests)
- Systemtests (testen ob Anforderungen erfüllt sind)

2 Begriffe, Abkürzungen und Referenzen

2.1 Begriffe & Abkürzungen

Abkürzung	Beschreibung
-----------	--------------

2.2 Referenzen

- keine

Testplanung

2.3 Unit-Tests

Zu möglichst jeder Klasse soll ein Unit-Test erstellt werden. Die Tests werden parallel zur Entwicklung erstellt und werden laufend ergänzt und angepasst. Der Fokus bei den Unit-Tests liegt vor allem bei den Kernklassen wie MastermindEngine, Settings, Guess, Result etc.

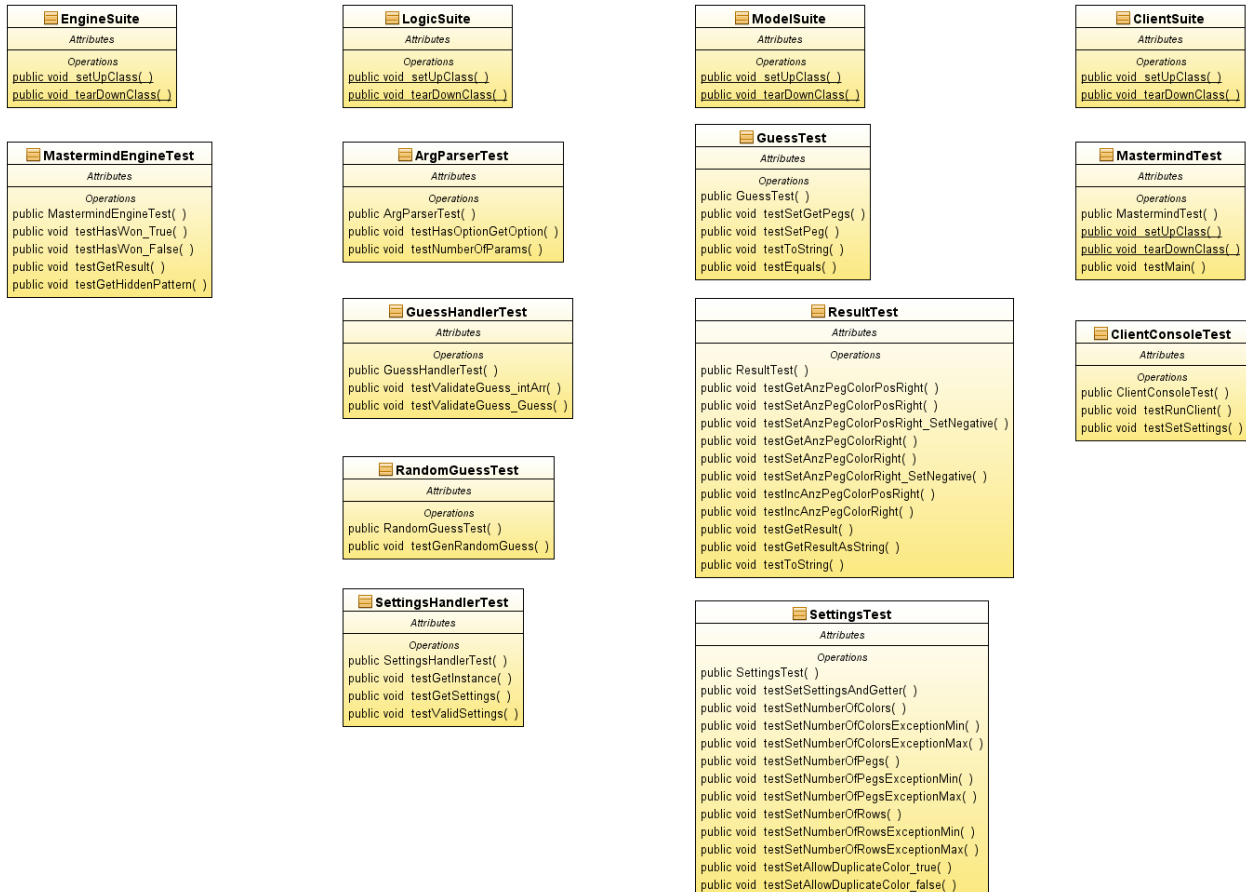


Abb. 1: JUnit Test Klassen

2.4 System-Tests

Systemtests werden anhand der Testfälle durchgeführt. Die System-Tests erfolgen durch die Projektmitglieder.

3 Test-Design

Unit-Tests werden innerhalb der Entwicklungsumgebung erstellt. Unit-Tests werden jeweils nach Bearbeitung resp. Änderungen im Code getestet. Wenn alle Unit-Tests bestanden wurden, werden die Systemtests durchgeführt und protokolliert.

4 Testfälle

# 1 Spiel starten / Rolle auswählen: Setzer / Rater	
Beschreibung	Der Spieler startet das Programm/Spiel
Akteure	Spieler
Auslöser	Der Spieler möchte ein Spiel spielen
Vorbedingungen	Computer (Tastatur & Maus), Java installiert
Ergebnis	Spiel wurde gestartet
Ergebnis im Fehlerfall	Spiel startet nicht
Ablauf	<ol style="list-style-type: none"> 1. Spielprogramm starten 2. Spielart auswählen (Optionen) 3. Spiel beginnen

# 2 Spiel speichern	
Beschreibung	Spielstand speichern
Akteure	Spieler
Auslöser	Der Spieler klickt auf Speichern
Vorbedingungen	Spiel muss gestartet sein
Ergebnis	Spielstand wurde gespeichert
Ergebnis im Fehlerfall	Spielstand kann nicht gespeichert werden und der Spielstand geht verloren.
Ablauf	<ol style="list-style-type: none"> 1. Speicher 2. Abfrage, wohin das File gespeichert werden soll 3. File wird gespeichert mit den nötigen Informationen

# 3 Spiel laden	
Beschreibung	Der Spieler möchte ein bereits gestartet Spiel laden
Akteure	Spieler
Auslöser	Spieler klickt auf laden
Vorbedingungen	File vorhanden mit dem aktuellen Spielstand
Ergebnis	Der letzte Spielstand kann geladen werden
Ergebnis im Fehlerfall	Das Spiel kann nicht geladen werden mit dem letzten Spielstand
Ablauf	<ol style="list-style-type: none"> 1. Spiel starten 2. Gespeichertes Spiel laden 3. Spiel spielen

4 Spiel beenden

Beschreibung	Der Spieler möchte das Spiel beenden
Akteure	Spieler
Auslöser	Spieler klickt auf Beenden
Vorbedingungen	Das Spiel muss gestartet sein
Ergebnis	Das Spiel wird beendet mit der Nachfrage, ob der Spielstand gespeichert werden soll
Ergebnis im Fehlerfall	Spiel kann nicht beendet werden
Ablauf	<ol style="list-style-type: none">1. Spiel beenden klicken2. Abfrage, ob evtl. der Spielstand gespeichert werden soll3. Spiel ist beendet

5 Spiel Hilfe anzeigen

Beschreibung	Der Spieler möchte die Hilfe anzeigen lassen
Akteure	Spieler
Auslöser	Spieler klickt auf Hilfe
Vorbedingungen	Spiel muss gestartet sein
Ergebnis	Die Hilfe wird angezeigt mit den nötigen Informationen
Ergebnis im Fehlerfall	Leeres Fenster wird geöffnet oder das Hilfe-Fenster erscheint nicht
Ablauf	<ol style="list-style-type: none">1. Spieler klickt auf Hilfe2. Hilfe-Fenster öffnet3. Hilfe-Fenster schliessen

6 About-Dialog anzeigen

Beschreibung	Der Spieler möchte das About-Dialogfenster anzeigen lassen
Akteure	Spieler
Auslöser	Spieler klickt auf About
Vorbedingungen	Spiel muss gestartet sein
Ergebnis	Das About-Dialogfenster erscheint mit den nötigen Informationen
Ergebnis im Fehlerfall	Das About-Dialogfenster erscheint nicht
Ablauf	<ol style="list-style-type: none">1. Spieler klickt auf About2. About-Dialogfenster wird angezeigt3. Spieler schliesst das About-Dialog

7 Optionen auswählen

Beschreibung	Der Spieler wählt seine Optionen für das nächste Spiel
Akteure	Spieler
Auslöser	Der Spieler startet das Spiel
Vorbedingungen	Das Spiel muss gestartet sein
Ergebnis	Die Optionen können ausgewählt werden und das Spiel wird dem entsprechen gestartet
Ergebnis im Fehlerfall	
Ablauf	<ol style="list-style-type: none"> 1. Spieler startet das Spiel 2. Spieler wählt seine Optionen 3. Spieler spielt das Spiel

8 Farbkombination eingeben (raten)

Beschreibung	Der Spieler hat das Spiel gestartet und gibt die mögliche Farbkombination ein
Akteure	Spieler
Auslöser	Spieler hat das Spiel gestartet und möchte die Farbkombination erraten
Vorbedingungen	Das Spiel muss gestartet sein
Ergebnis	Die Farbkombinationen können eingegeben werden und das Spiel entnimmt diese Eingabe
Ergebnis im Fehlerfall	
Ablauf	<ol style="list-style-type: none"> 1. Spieler hat das Spiel gestartet 2. Spieler gibt die mögliche Farbkombination an

9 Farbkombination vergleichen/bewerten (Vorgabe vs. gesetzt)

Beschreibung	Der Spieler möchte seine eingegebenen Farbkombinationen vergleichen
Akteure	Spieler
Auslöser	Der Spieler möchte seine Eingabe überprüfen
Vorbedingungen	Das Spiel muss gestartet sein und die Farbkombinationen müssen erfasst sein
Ergebnis	Das Spiel vergleicht die Eingabe des Spielers
Ergebnis im Fehlerfall	
Ablauf	<ol style="list-style-type: none"> 1. Spieler startet Spiel 2. Spieler gibt die Farbkombinationen ein 3. Spieler klickt auf überprüfen

# 10 Spiel gewonnen / verloren	
Beschreibung	Spieler möchte wissen, ob er das Spiel gewonnen oder verloren hat
Akteure	Spieler
Auslöser	Spieler klickt auf überprüfen
Vorbedingungen	Eine Farbkombination muss erfasst sein
Ergebnis	Das Spielt zeigt an, ob der Spieler gewonnen oder verloren hat
Ergebnis im Fehlerfall	
Ablauf	<ol style="list-style-type: none">1. Spieler startet das Spiel2. Spieler gibt seine Farbkombination ein3. Spieler klickt auf überprüfen4. Der Spieler wird informiert, ob er das Spiel gewonnen oder verloren hat

Abbildungsverzeichnis

Abb. 1: JUnit Test Klassen..... 2