



Mastermind

Kundenanforderungen

**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, 03.05.2012

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	21.04.2012	Felix Rohrer	Vorlage erstellen, div. Texte erstellen
	22.04.2012	Marco Reichmuth	Produktübersicht
	24.04.2012	Iljasa Redzepi	Use Case beschreiben
	25.04.2012	Philipp Rey	Produktdaten, Qualitätsanforderungen
	03.05.2012	Philipp Rey	Korrektur Rechtschreibung
1.1	03.05.2012	Felix Rohrer	Version 1.1 freigeben

Inhalt

1	Ziel und Zweck.....	1
1.1	Ziel	1
1.2	Zweck.....	1
2	Begriffe, Abkürzungen und Referenzen	1
2.1	Begriffe und Abkürzungen.....	1
2.2	Referenzen	1
3	Produktübersicht.....	2
3.1	Beschreibung über das Produkt	2
3.2	Spielregeln	2
4	Produkt-Funktionen	4
4.1	Funktionen / Use-Cases.....	4
5	Produktdaten	8
6	Qualitätsanforderungen, Leistungsanforderungen	8
6.1	Aus Anwendersicht.....	8
6.2	Aus Entwicklersicht	8
7	Ergänzungen.....	9
	Abbildungsverzeichnis.....	9

1 Ziel und Zweck

1.1 Ziel

Als interdisziplinäre Projektarbeit im Modul Programmieren 2 an der Hochschule Luzern Technik & Architektur lautet der Projektauftrag ein Computerspiel „Mastermind“ zu programmieren.

Wahlweise kann der Spieler die Rolle des „Setzers“, „Bewerter“ (Codemaker / Codierer) oder „Raters“ (Codebreaker) übernehmen. Beide Rollen können wahlweise vom Computer übernommen werden. Via GUI (grafisches User Interface) kann das Spiel gespielt werden. Dadurch ist ein attraktives Spielvergnügen und Benutzerfreundlichkeit sichergestellt. Der aktuelle Spielstand lässt sich speichern und später wieder laden.

1.2 Zweck

Das Spiel ist primär als Unterhaltung gedacht. Bei der Erstellung des Programms sammeln die Studenten zusätzlich Erfahrung im Bereich Software Engineering.

2 Begriffe, Abkürzungen und Referenzen

2.1 Begriffe und Abkürzungen

Abkürzung	Beschreibung
Bewerter	Bewertet die Farbfolge des Raters (und ist gleichzeitig der Setzer).
Codebreaker	siehe Rater
Codemaker	siehe Setzer / Bewerter
GUI	„Graphical User Interface“, Grafisches User Interface Grafische Benutzeroberfläche
Rater	Versucht die Farbfolge des Setzers zu erraten.
Setzer	Gibt die zu erratende Farbfolge vor und ist gleichzeitig der Bewerter.

2.2 Referenzen

- Auftrag Softwareprojekt FS 2012 (PRG2_PRJ_Auftrag_2012-V1.1.pdf)
- MasterMind Automatischer Solver (PRG2_PRJ_MasterMind_Automat.pdf)

3 Produktübersicht

3.1 Beschreibung über das Produkt

Bei diesem Produkt handelt es sich um ein Spiel, welches am PC gespielt werden kann. Es ist möglich 2 verschiedene Rollen einzunehmen. Zum einen kann man den zu erratenden Code setzen und die Auswertung machen oder man errät den Code. Spielgegner ist eine weitere Person oder der PC.

Die Anzeige ist über ein GUI realisiert. Eingaben können mit der Maus gemacht werden. Das Spiel verfügt über verschiedene Optionen. Anhand dieser kann z.B. die Schwierigkeit eingestellt werden. Das Spiel kann jederzeit gespeichert werden und entsprechend auch wieder geladen werden.

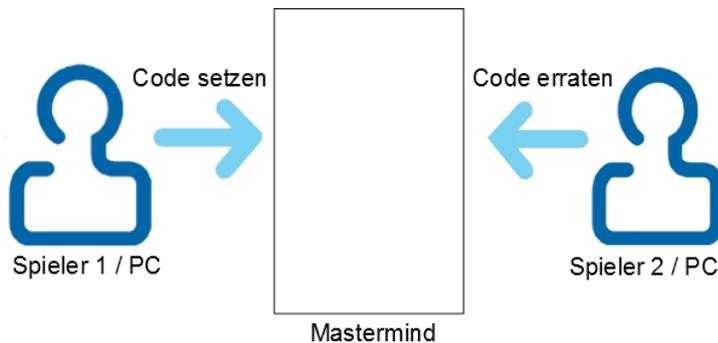


Abb. 1 Mastermind: Code setzen / erraten

3.2 Spielregeln

3.2.1 Allgemeines

Mastermind ist ein Spiel für zwei Spieler. Ein Spieler gibt eine Kombination von Farben vor, und der andere versucht diese in einer vorgegebenen Anzahl von Versuchen zu erraten. Der Vorgebende bewertet dabei jeden Versuch des Ratenden nach seiner Korrektheit mit schwarzen und weissen Punkten.

3.2.2 Bewertungsregeln

- Schwarzer Punkt: richtige Farbe an der richtigen Stelle erraten
- Weisser Punkt: richtige Farbe aber an die falsche Stelle gesetzt

Dabei hat die Reihenfolge der gegebenen Punkte nichts mit der Reihenfolge in der geratenen oder der vorgegebenen Kombination zu tun. Jede Farbe in der Vorgabe und im Versuch darf nur einmal zur Bewertung herangezogen werden.

Beispiel:

Kombination	●	●	●	●	●
Versuch	●	●	●	●	●
Punkte	●	○	○	●	●

Abb. 2 Bewertungsregeln

3.2.3 Optionen

Die Spieloptionen können mit diesen Parametern gewählt werden:

- Farben: 4-8
- Länge der Kombination: 4-6
- maximale Versuchsanzahl: 6-12
- mehrfache Farben erlaubt: Ja/Nein

3.2.4 Computer gibt vor

Der Computer gibt eine zufällige Kombination mit den gewünschten Spezifikationen vor. Danach kann der Spieler seine Versuche starten, und diese vom Computer bewerten lassen.

3.2.5 Spieler gibt vor

Der Spieler gibt eine Kombination vor, oder lässt sich eine zufällig erzeugen. Der Computer rät eine Kombination, und der Spieler bewertet diese dann. Bei Falscher Bewertung kommt es zu einem Fehler.

3.2.6 Hilfen

- Ratehilfe: Der Computer gibt aus den noch möglichen Kombinationen zufällig eine vor.
- Hilfe: Der Computer verrät die Farbe an der gewünschten Position.
- Lösen: Der Computer gibt die Kombination aus.

Aus den Optionen und den benötigten Versuchen wird dann eine Punktzahl errechnet.

4 Produkt-Funktionen

4.1 Funktionen / Use-Cases

# 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	
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 Spiel2. Spieler wählt seine Optionen3. 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 gestartet2. 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 Spiel2. Spieler gibt die Farbkombinationen ein3. 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

5 Produktdaten

Die Anforderung, die das Spiel an das System zur Ausführung stellt, sind eine installierte Java-Laufzeitumgebung (JRE), eine Maus zur User-Eingabe und ein vorhandenes Netzwerk, welches eine Verbindung zu einem anderen System sicherstellt, um Spieler ortonabhängig gegeneinander antreten zu lassen.

6 Qualitätsanforderungen, Leistungsanforderungen

6.1 Aus Anwendersicht

Bei der Ausführung der Spiels sollen keine Exceptions/Errors auftreten und jegliche Eingaben müssen sofort ohne Verzögerung umgesetzt werden (Performance). Ressourcen des Systems sollen nicht unnötig ausgelastet werden, um das System nicht zu verlangsamen oder anderweitig zu beeinträchtigen.

Die Oberfläche des Spiels soll intuitiv und ansprechend gestaltet sein, so dass ein Spieler, welcher mit den Spielregeln vertraut ist, sofort mit dem Spiel beginnen kann. Dasselbe gilt auch für Zusatzfunktionen wie Spielstände speichern und wieder abrufen.

6.2 Aus Entwicklersicht

Der Code des Spiels soll so aufgebaut sein, dass er folgende Punkte zulässt:

- Erweiterbarkeit
- Wartbarkeit
- Übertragbarkeit
- Wiederverwendbarkeit

7 Ergänzungen

Eine erste Version des Spiels besitzt noch keine GUI, spätere Versionen werden nur noch via GUI spielbar sein. Verschiedene Erweiterungen sind Denkbar, diese müssen jedoch explizit zusätzlich spezifiziert werden. Einzig die Erweiterung, dass das Spiel via Netzwerk von zwei Computern aus gespielt werden kann, wird je nach Verfügbarer Zeit umgesetzt. Ausgabe von Tönen / Musik ist nicht geplant.

Abbildungsverzeichnis

Abb. 1 Mastermind: Code setzen / erraten.....	2
Abb. 2 Bewertungsregeln.....	2