**Dokumentation Projekt M318**

*Brettspiel Dame (in c# programmiert)*

Verfasser: Fabio Litscher

Version: 1.0

Erstellungsdatum: 02.06.2014

**Inhalt**

[1. Einleitung 3](#_Toc389487131)

[2. Benutzerschnittstelle 4](#_Toc389487132)

[2.1. Aufgabenangemessenheit 4](#_Toc389487133)

[2.2. Selbstbeschreibungsfähigkeit 4](#_Toc389487134)

[2.3. Steuerbarkeit 4](#_Toc389487135)

[2.4. Erwartungskonformität 4](#_Toc389487136)

[2.5. Fehlertoleranz 4](#_Toc389487137)

[2.6. Individualisierbarkeit 4](#_Toc389487138)

[2.7. Lernförderlichkeit 4](#_Toc389487139)

[3. Fehlermeldungen 5](#_Toc389487140)

[4. Benutzergruppe 6](#_Toc389487141)

[4.1. Beurteilung entsprechend Benutzermodell 6](#_Toc389487142)

[4.2. Nötige Anpassungen / Änderungen 6](#_Toc389487143)

[5. Zustandsdiagramm 7](#_Toc389487144)

[6. Codekonventionen 8](#_Toc389487145)

[7. Wartung 9](#_Toc389487146)

[8. Programm Dokumentation 10](#_Toc389487147)

[9. Code Review 11](#_Toc389487148)

# Einleitung

In diesem Projekt, im Rahmen des Modules 318, geht es darum das Brettspiel Dame in der Sprache C# zu programmieren.

Das Ziel ist, dass man mindestens Spieler gegen Spieler spielen kann, ein Zusatz wäre, wenn man Spieler gegen Computer spielen könnte.

Da ich nach drei Wochen bereits alleine in der Gruppe war, da meinem Partner der Lehrvertrag aufgelöst wurde, wurde das Projekt dann von mir alleine realisiert.

Das Spiel Dame ist ein klassisches Brettspiel, welches von jeder Altersgruppe gespielt werden kann. Damit die Bedienung möglichst einfach ist, habe ich mich entschieden, das ganze möglichst schlicht darzustellen und mit wenigen intuitiven Knöpfen die wichtigsten Funktionalitäten zu gewährleisten.

# Benutzerschnittstelle

## Aufgabenangemessenheit

**Gut:**

Unnötige Interaktionen wurden vermieden, indem beispielsweise am Anfang bereits feststeht, dass immer Weiss beginnt.

Standardwerte wie beispielsweise „Spieler 1“ und „Spieler 2“ sind bereits gegeben.

## Selbstbeschreibungsfähigkeit

**Durchschnittlich:**

Hilfen wurden keine grossen eingebaut, allerdings ist das Programm sehr verständlich aufgebaut, da es wenige Knöpfe gibt, bei denen man aber sofort weiss für was sie stehen.

## Steuerbarkeit

**Durchschnittlich:**

Da auf Zusatzfeatures verzichtet wurde, und vor Allem die Grundfunktionalität (Spieler vs. Spieler) gewährleitet wird, gibt es keine grossen Einstellungsmöglichkeiten, welche der Benutzer steuern könnte.

## Erwartungskonformität

**Durchschnittlich:**

Der Grossteil der Knöpfe ist selbsterklärend. Das heisst der Benutzer weiss immer was er macht, und macht nicht ausversehen etwas, was er eigentlich nicht wollte.

## Fehlertoleranz

**Durchschnittlich:**

Es existieren nicht viele Fehlerquellen im Programm, da nicht viele Eingabefelder existieren.

Was allerdings überprüft wird, ist ob der Spieler nach Regeln spielt, man kann also keinen Stein auf ein Feld setzen, wenn man das laut Regelwerk nicht darf.

## Individualisierbarkeit

**Weniger berücksichtigt:**

Der Spieler hat keine grossen Anpassungsmöglichkeiten, die Werte sind vorgegeben.  
Es wurde weniger Wert darauf gelegt, da an erster Stelle die Funktionalität stand.

## Lernförderlichkeit

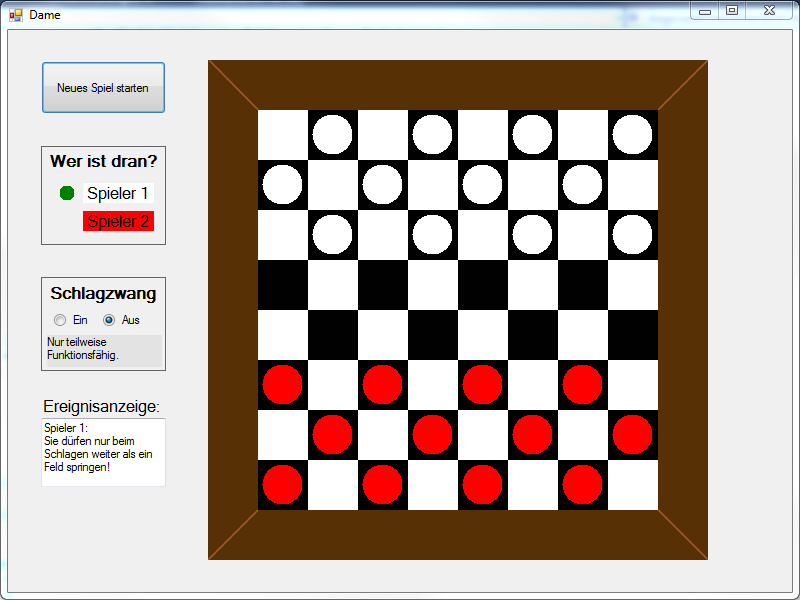
**Gut:**

Das Programm ist so aufgebaut, dass auch Neuanwender keine Probleme haben das Programm zu bedienen.

# Fehlermeldungen

Fehlermeldungen, wurden bei meinem Programm praktisch keine implementiert, bis auf diejenigen, die in Einsatz treten, wenn man einen unzulässigen Spielzug durchführen möchte. Diese werden jeweils links unten neben dem Spielfeld in der Ereignis-Konsole angezeigt.

Beispielsweise kommt folgende Meldung, wenn der Spieler 1 mit einem seiner weissen Steine, weiter als ein Feld fahren möchte, ohne dass er einen gegnerischen Stein schlagen kann:



# Benutzergruppe

## Beurteilung entsprechend Benutzermodell

Laut den Definitionen der drei verschiedenen Benutzergruppen, entsprechend dem Benutzermodell (Ergonomie\_BalzertLE16.pdf), würde ich mein Programm folgendermassen beurteilen:

|  |  |  |
| --- | --- | --- |
| **Anfänger** | **Gelegenheitsbenutzer** | **Experten** |
| * Einfache Bedienung durch wenige, dafür klare Funktionen (Knöpfe). * Keine grossen Einstellungsmöglichkeiten. | * Bedienung wie bei anderen gewohnten Programmen. * Keine Schwierigkeiten um zu verstehen, was passiert, wenn man beispielsweise auf einen Knopf klickt. | * Wenn man weitere Funktionen implementieren möchte, ist das eine gute Basis. * Überhaupt keine Probleme bei der Bedienung, findet sich blind zurecht. |
| * Keine Hilfe mit einer ausführlichen Erklärung zum Programm. * Keine interaktiven Hilfestellungen. | * Funktionsumfang evtl. etwas knapp, da kein Modus Spieler vs. Computer möglich ist. | * Geringer Funktionsumfang. * Wenige Einstellungs-möglichkeiten. |

Gemäss dieser Bewertung, würde ich sagen, ist mein Programm hauptsächlich für Anfänger und Gelegenheitsbenutzer entworfen.

## Nötige Anpassungen / Änderungen

Um den Experten-Bereich besser abzudecken, wären folgende Zusätze / Anpassungen nötig:

* Grösserer Funktionsumfang
  + Spieler vs. Computer
  + Speichern von Highscore
  + Spielstand in Datei speichern, Spielstand wieder aus Datei laden
* Mehr Einstellungs- / Personalisierungsmöglichkeiten
  + Möglichkeit zum Anpassen des Farbschemas
  + Skalierbarkeit

Folgende Sachen wären mit geringem Aufwand möglich zu implementieren:

* Anpassung der Farbe der Spielsteine
  + Mit Hilfe einer kleinen Auswahl aus 4 - 5 fix festgelegten Farben.
* Skalierbarkeit
  + Wollten wir anfangs umsetzen, dann aber auf Eis gelegt, alles auskommentiert.
* Spielstand in Datei speichern
  + Alle Array Daten in ein Textdokument ausgeben.

# Zustandsdiagramm

# Codekonventionen

# Wartung

# Programm Dokumentation

# Code Review