Pascal Honegger  
Pflichtenheft „Chess“

15.09.2015

# Übersicht

## Projekthintergrund und Beschreibung

Die Grundidee für das Schach entstand in einer gewöhnlichen Konversation mit meinem Freund Alain Keller. Dabei erwähnte er, dass er ein Schach programmieren wolle. Später besprachen wir Details bezüglich Programmiersprache, Arbeitsaufteilung und Funktionen des Schachs. Wir kamen zum Schluss, dass wir das Schach in C# programmieren möchten und wir zur Versionskontrolle Git verwenden, da ich viel Erfahrung mit beiden Tools besitze. Das Endprodukt „Schach“ sollte alle Schachregeln beherrschen, genaueres unten.

## Projektumfang

Dieses Projekt umfasst ein vollumfängliches Schach (unten definiert). Das Schach ist mit Sinnvollen Unit-Tests deckend getestet. Das Schach kann über einen mitgelieferten Installer installiert werden.

## Anforderungen auf hoher Ebene

Das Finale Schach muss Folgende Funktionen besitzen:

* Es kann eine andere Textur für das Schachbrett gewählt werden (Drop-Down)
* Das Spiel kann über einen Start-Knopf gestartet und neu-gestartet werden
* Es muss Schach gespielt werden können:
  + Figuren werden beim Klicken markiert. Alle möglichen Züge werden grün (Gehen) und orange (Essen)
  + Bei erneutem Klicken wird, falls ein möglicher Weg vorhanden ist, die Figur zum Zielpunkt bewegt
  + Die gefressene Figur wird in dem Friedhof bewegt
  + Der Spielverlauf wird als Text dargestellt (Bsp. A3 => B6)
  + [Freiwillige Erweiterung] Der Spielverlauf wird Grafisch dargestellt, wobei man A3 => B6 als Texturen und Text sehen kann
* Die Spielzeit jedes Spielers wird gemessen
  + Falls ein Unentschieden entsteht, gewinnt der Spieler mit der kürzeren Gesamt-Rundenzeit
* Jeder Spieler kann seinen Benutzernahmen wählen
  + Der Benutzername ist maximal 30 Zeichen lang und unterstützt Fremdzeichen
* [NUnit Tests] Alle Funktionen des Schachs werden per automatisierten Tests auf Fehler überprüft.
* [Blackbox Tests] Es sind Testfälle für ein grundlegendes Schachspiel definiert, welches alle Figuren, die Rochade und das Schachmatt beinhalten

## Lieferumfang

Voll funktionsfähiges, den Anforderungen entsprechendes Schach inklusive Installationsdatei, Quellcode (Unit Tests), Klassendiagramm, einem grösseren Testfall und einer kurzen (2 Seiten Text) Dokumentation.

## Implementierungsplan

* Verwendete Sprachen
  + C#
  + XAML (WPF Applikation)
* Programmierumgebung
  + Visual Studio Enterprise 2015
    - NUnit
    - ReSharper Ultimate 2015
    - (MVVM-Prinzip)
  + Laptop
    - 8GB Ram
    - 4 Cores @ 1.8 GHZ
    - Windows 10 Pro

## Zeitskala/Planung auf hoher Ebene

Da bereits vor der Erstellung des Pflichtenheftes an dem Projekt gearbeitet wurde, beginnt der Zeitplan nicht beim Erstellen des Programmes.

|  |  |
| --- | --- |
| Datum | Ziel |
| 29.09.2015 | Abklärung, welche Kompetenzen mit dem Schach erfüllt werden können |
| 20.10.2015 | Fertigstellung NUnit-Tests |
| 27.10.2015 | Fertigstellung GUI-Implementation, Schach spielen begrenzt möglich |
| 03.11.2015 | Implementation Schach-Matt |
| 10.11.2015 | Implementation Rochade |
| 17.11.2015 | Code Review, Abschliessen Projekt. |
| 24.11.2015 | Know-How-Transfer an Alain Keller  Kompetenzen bestätigen lassen |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **André Frey** |  |  |  |  |  |  |  |  |
| Genehmigt von |  |  | Datum |  | Genehmigt von |  |  | Datum |