

# Angabe für das Schlussprojekt in „Programmieren“

*Bei diesem Dokument handelt es sich um die Originalangabe für Gruppe 2 von Hr. Markus Safar. Ich verwende es auch für Gruppe 3 und ändere es an gegebenen Stellen (in blau dargestellt); dabei werde ich Erweiterungen angeben bzw. nicht relevante Dinge durchstreichen. Damit soll die nötige Transparenz gegeben sein, um später einen gemeinsamen Präsentations- und Testtag durchführen zu können. Danke Hr. Safar für die Definition der Angabe!*

*(Martin Mayr, Jänner 2013)*

## 1 Einführung

Das vorliegende Dokument beschreibt die Aufgabenstellung<sup>1</sup> für das Abschlussprojekt. Es handelt sich um eine 3D-Variante des klassischen Brettspiels *4 Gewinnt*<sup>2</sup>. Der Regelsatz des Spiels bleibt unverändert, die Spieler setzen abwechselnd einen Stein, jener Spieler, der zuerst 4 Steine auf eine Linie<sup>3</sup> legt, gewinnt das Spiel.

## 2 Anforderungen

Nachfolgend werden die **Mindestanforderungen** beschrieben. Die Nichteinhaltung dieser Mindestanforderungen resultiert automatisch in einer negativen Note.

~~Das Programm muss aus Enums, Klassen und Interfaces bestehen.~~ Das Programm muss vollständig objektorientiert programmiert sein – das Klassendiagramm ist bei der Abschlusspräsentation zu präsentieren. Die Klassen und deren Methoden müssen sinngemäß gekapselt werden, so dürfen in Methoden die z.B.: der Berechnung von Werten dienen keine Eingaben von bzw. Ausgaben am Bildschirm vorgenommen werden. Entwickeln Sie das Programm in Hinblick auf Erweiterbarkeit.

Für Gruppe 2 gilt: ALLE StyleCop-Regeln sind einzuhalten.

### 2.1 Bedienbarkeit

Das Programm muss für den Benutzer intuitiv bedienbar sein, Fehleingaben müssen so weit wie möglich toleriert werden. Abstürze sämtlicher Art sind zu vermeiden. Das bedeutet, dass sämtliche Laufzeitfehler abgefangen, entsprechend behandelt und ggf. dem Benutzer in sinnvoller Art und Weise zum Quittieren übergeben werden müssen.

<sup>1</sup> Allgemeiner Hinweis zur Abgabe: Es wird eine Abgabe inklusive Abgabe-Prüfungsgespräch geführt in dem nachgewiesen werden muss, dass sämtliche Inhalte selbstständig erarbeitet und verstanden wurden. Dies wird unter Umständen auch durch implementieren von Funktionalität während des Prüfungsgesprächs überprüft.

<sup>2</sup> Details zum Regelwerk können unter [http://de.wikipedia.org/wiki/Vier\\_gewinnt](http://de.wikipedia.org/wiki/Vier_gewinnt) nachgelesen werden.

<sup>3</sup> Da es sich hier um eine 3D-Variante des Spiels handelt, sind auch alle im dreidimensionalen Raum liegenden Steine, die eine Linie bilden, zu berücksichtigen.

## 2.2 Spielmodi

Der Benutzer kann vor Beginn eines Spiels einen von 3 Spielmodi wählen. Der Spielmodus kann erst wieder nach Ende eines Spiels und vor Beginn eines neuen Spiels geändert werden.

Die zur Verfügung gestellten Spielmodi sind

- *Mensch gegen Mensch,*
- *Mensch gegen Maschine und*
- *Maschine gegen Maschine.*

Beachten Sie, dass bei den Modi \* gegen Maschine immer gegen die KI auf einem anderen System gespielt werden muss, dies ist also nur durch die Auswahl eines Spiels, dass bereits im Laufen ist möglich (dadurch wird automatisch ein Computergegner erstellt).

Es muss für den Computerspieler ein Delay (einzugeben in der GUI – Default: 1000ms) einzustellen sein. Dadurch wird ermöglicht, dass bei Maschine gegen Maschine das Spiel visuell verfolgt werden kann.

## 2.3 Spielfeld

Die Dimension des Spielfelds kann beliebig gewählt werden, natürlich müssen immer genügend Steine zur Verfügung stehen um alle Felder füllen zu können. Zu beachten ist, dass die Anzahl der Spielsteine für jeden Spielteilnehmer ident sein muss. Ein Spielfeld in der Größe 7 Spalte x 7 Zeilen x 1 Ebene ist daher nicht zulässig. Die Spielfeldgröße muss so gewählt werden, dass ein sinnvolles spielen des Spiels möglich ist. ~~Da es sich um eine 3D-Variante handelt macht ein Spielfeld mit < 4 Ebenen in die Tiefe keinen Sinn.~~

## 2.4 Architektur

Die Architektur der Software besteht im Wesentlichen aus Clients und einem oder mehreren Indizierungsservern. In Clients werden beliebig viele Spiele gehostet. Diese werden anschließend über Indizierungsserver bekannt und zugänglich gemacht. Es können beliebig viele Spielprozesse pro Host gestartet werden, jeder Spielprozess kann beliebig viele Spiele zur Verfügung stellen (wenn ein menschlicher Spieler im Spiel ist und ein Client sich zu dem gehosteten Spiel connected wird automatisch ein Computerspieler erstellt).

## 3 Netzwerkprotokoll

Das Netzwerkprotokoll inklusive einiger Szenarien, die den detaillierten Ablauf darstellen, sind in der Datei *Netzwerkprotokoll.pdf* hinreichend beschrieben. Diese Angaben sind ausnahmslos einzuhalten.

## 4 Benotung und Präsentation

Das Projekt wird bei einem persönlichen Abschlussprojekt präsentiert. Dabei sind mindestens folgende Teile zu präsentieren:

- Konzeptueller Entwurf (Skizzen)
- Klassendiagramm vor Implementierung
- Fertiges Klassendiagramm
- Alle „Spezialinhalte“ zur Notenverbesserung (siehe nächster Punkt)

Generell muss die Funktionalität des Spiels gewährleistet sein, das bedeutet:

- die netzwerktechnische Umsetzung (Implementierung des Protokolls),
- die Implementierung der 3 Spielmodi
- eine geeignete Visualisierung

Zusätzlich können Sie sich „Bonuspunkte“ erarbeiten – um sich die Note um jeweils einen Grad verbessern zu können. Dazu stehen folgende 3 Punkte zur Verfügung:

- 3D Visualisierung des Spiels (z.B. via OpenGL)
- KI: Aktive Gewinnstrategie (also nicht nur verteidigen – das ist Voraussetzung, sondern gewinnorientiertes Setzen von Steinen). Das muss beim Abschlussgespräch entsprechend dargestellt werden.
- Android Implementierung für Smartphones/Tabs mit Hilfe von „necessitas“.

Termine bzw. Details zum Präsentationstag folgen...

Bei Fragen stehe ich gerne zur Verfügung!

Martin Mayr, Jänner 2013

## **5 – Mindestanforderungen pro Notengrad und Überprüfung**

~~Nachfolgend werden die jeweiligen Mindestanforderungen pro Notengrad beschrieben. Die Anforderungen setzen aufeinander auf, so sind z.B.: zum Erreichen der Note „Befriedigend“ alle Anforderungen für „Genügend“ und zusätzlich die Anforderungen für die Note „Befriedigend“ zu erfüllen.~~

### **5.1 – Genügend**

~~Sämtliche Funktionalitäten des Clients (aus Sichtweise des Gegenspielers) müssen basierend auf der Protokoll-Version 1 implementiert werden. Dies beinhaltet unter anderem die Suche nach verfügbaren Spielen, das Verbinden zu einem verfügbaren Spiel sowie die Teilnahme an einem Spiel als Gegenspieler. Die Visualisierung kann wahlweise als Consolen-Applikation (bevorzugt) oder WPF-Applikation realisiert werden.~~

### **5.2 – Befriedigend**

~~Sämtliche Funktionalitäten des Clients (aus Sichtweise des Gegenspielers) und des Servers (stellt ein Spiel zur Verfügung) müssen basierend auf Protokoll-Version 2 implementiert werden. Zusätzlich ist die Implementierung des Indexservers notwendig. Dieser muss als eigenständige Console-Applikation realisiert werden. Achten Sie darauf, dass die Implementierung des Servers eine entsprechende künstliche Intelligenz erfordert (Computergegenspieler). Um diesen Notengrad zu erlangen ist die Implementierung einer passiven KI (versucht ein Verlieren des Spiels zu verhindern) ausreichend.~~

### **5.3 – Gut**

~~Um diesen Notengrad zu erreichen kann wahlweise eine aktive KI (versucht das Spiel aktiv zu gewinnen) oder eine entsprechen ansprechende Visualisierung implementiert werden.~~

~~Beachten Sie aber, dass das Userinterface trotz ansprechender Visualisierung nach wie vor intuitiv bedienbar sein muss.~~

## **5.4 – Sehr Gut**

~~Um diesen Notengrad zu erreichen muss der offen gelassene Punkt des Notengrads „Gut“ implementiert werden. Haben Sie also bereits ein entsprechend ansprechendes Userinterface implementiert so müssen Sie nun eine aktive KI implementieren. Sollten Sie für den Notengrad „Gut“ bereits die aktive KI gewählt und implementiert haben so bleibt Ihnen nun die Implementierung eines ansprechenden Userinterfaces zur Wahl.~~

## **5.5 – Überprüfung**

~~Es wird eine Abgabe inklusive Abgabe-Prüfungsgespräch geführt in dem nachgewiesen werden muss, dass sämtliche Inhalte selbstständig erarbeitet und verstanden wurden. Dies wird unter Umständen auch durch implementieren von Funktionalität während des Prüfungsgesprächs überprüft.~~