

## 3. Abgabe Software-Praktikum 2011

Mit dieser Abgabe erzeugt ihr Version 0.3 eures Programms.

In dieser Version implementiert ihr die Netzwerk-Protokoll und die Künstliche-Intelligenz. Wenn eure vorherigen Abgabe nicht komplett waren, musst ihr sie auch fertig machen.

### Abgabe:

- Bis Dienstag 24. Mai, 18:00, ins SVN-Repository eurer Gruppe.
- Pfad: In einem Branch „version-0.3“ (ohne Anführungszeichen, wie von nun an), das heißt, im Verzeichnis „branches/version-0.3“. Innerhalb dieses Ordners musst eure Quelltext unter „src“ stattfinden: eure Applikation unter „src/main“, eure JUnit-Testen unter „src/test“. Das heißt, eine JUnit-Test für das Spielbrett wird wahrscheinlich heißen „branches/version-0.3/src/test/GameboardTest.java“.

### 1 Austauschbare Text-Testfälle erzeugen und testen

Das war schon letztes mal erfordert, aber niemand hat sie erzeugt.

Schreibt vorschriftsmäßig Testfälle-Dateien für eure Implementation (Eingabe und erwartete Ausgabe). Jeder Testfall umfasst eine Sequenz von Zügen, das heißt ein Board (welches in der Beschreibung des Netzwerk-Protokolls angegeben ist). Für jeden Zug wird eine Zeile ausgegeben, die den neuen Punktestand zeigt, wenn der Zug gültig ist. Die Zeile richtet sich nach Format `Option[Score]` (auch siehe Netzwerk-Protokolls Beschreibung).

Ihr sollt auch ein kleines Test-Framework schreiben, dass die Eingabe und erwartete Ausgabe liest und dann bestätigt, ob eure Implementation die korrekte Ausgabe erzeugt. Dann sollt ihr mit diesem Framework Testfälle [von mir](#) und von euch (demselben Muster entsprechend) testen. Dieses Test-Framework kann als ein JUnit-Test implementiert werden; auf jeden Fall muss es die Ergebnisse ausgeben. Ihr sollt Testfällen schreiben, so dass die besondere Schwierigkeiten (z.B. Grenzfälle wie Arraygrenzen) für eure Implementation auch getestet sind.

Diese sind Integrations-Testfälle, weil sie die komplette Spiellogik testen. Andere JUnit-Testfälle testen individuelle Methoden. Beide Arten von Testfälle sind wichtig.

### Abgabe:

- Text-Dateien: Ins Ordner `tests` unter den Branch `version-0.3`. Jeder Testfall besteht aus `inputN.txt` und `outputN.txt`, z.B. „input5.txt“ und „output5.txt“. Der komplette Pfad ist dann „branches/version-0.3/tests/input5.txt“.
- Test-Framework: mit eure Quelltext. Beschreibt ihr im „branches/version-0.3/README“ wie man den Framework benutzt.

### 2 Netzwerkkommunikation Unterstützung

Ihr Client sollt den Netzwerkprotokoll implementieren, damit ihre Client gegen andere Spieler spielen kann. Als Test, ihr sollt gegen meine KI spielen. `NetzwerkProtokoll.pdf` beschreibt

den Netzwerkprotokoll.

### **3 Künstliche Intelligenz**

Ihr sollt schreiben eine Künstliche Intelligenz (KI), die gegen meine dumme KI gewinnt. Meine dumme KI spielt immer auf den Test-Server. Der Test-Server ist erreichbar auf Port 34567 vom `vh12009.mathematik.uni-marburg.de`, wie erklärt im `NetzwerkProtokoll.pdf`. Die dumme KI heißt `StupidPlayer` („dumme Spieler“). Im Test-Server, melden sie an Spiel „`StupidGame`“ um gegen sie zu spielen.