

4. Abgabe, Zwischenabnahme Software-Praktikum 2011

Mit der 4. Abgabe entwickelt ihr Version 0.4 eures Programms.

In dieser Version fangt ihr die Implementierung der Graphische Benutzeroberfläche an, und ihr implementiert das Testprogramm für eure KI.

Abgabe:

- Bis Dienstag 14. Juni, 18:00, ins SVN-Repository eurer Gruppe.
Der Termin für die **Prüfungsanmeldung** ist am 17. Juni; ich werde euch vorher Feedback geben.
Wenn ihr verspätet abgibt, kann ich euch nicht garantieren, Feedback vor der Prüfungsanmeldung zu geben.
- Pfad: In einem Branch „version-0.4“ (ohne Anführungszeichen, wie von nun an), das heißt, im Verzeichnis „branches/version-0.4“. Euer Quelltext muss wie letztes Mal unter „src/main“ und JUnit-Tests unter „src/test“ stehen.

1 Graphische Benutzeroberfläche (GUI)

Die GUI soll soweit funktionieren, dass man über Netzwerk gegen KI oder andere Spieler (lokale oder remote) spielen kann. Dabei müssen alle regelgemäßen Züge möglich sein (also z.B. erste-Zug-Regel implementiert sein, Bonus-Züge, alle Spielsteine austauschen, usw.).

Die GUI soll die aktuellen Punktestände aller Spieler zeigen und die verfügbaren Steine des lokalen (KI oder menschlichen) Spielers.

2 Testprogramm für eure Künstliche Intelligenz (KI)

Damit ich eure KI gegen meine dumme KI testen kann, sollt ihr ein Testprogramm entwickeln. Dann kann ich das Testen automatisieren.

Die KI muss sich über Konsole starten lassen (über Argumente der Main-Methode). Sie muss sich an den Server (über Port 34568) verbinden, dann ein Spiel erzeugen oder sich an einem bestehenden Spiel anmelden, und dann spielen.

Derjenige, wer die KI startet, entscheidet, ob sie sich an einen bestimmte Spiel anmelden soll, oder ob sie ein Spiel mit einem bestimmten Namen und einer bestimmte Anzahl von Spielern erzeugen muss.

Ihre Main-Methode bekommt als Argumente den Name des Spielers, dann entweder JOIN oder CREATE, dann ein Spiel-Name, und am Ende (beim CREATE) ein Anzahl von Mitspielern.

Wenn die Classpath „classes“ ist und die Name von die Main-Klass „de.unimarburg.einfachGenial.MainKI“, werde ich dann laufen lassen:

```
java -cp classes de.unimarburg.einfachGenial.MainKI SpielerName  
JOIN GameName  
java -cp classes de.unimarburg.einfachGenial.MainKI SpielerName  
CREATE GameName nParticipants
```

Z.B., um zu testen, als Gruppe 3 zu spielen gegen die dumme KI „StupidGame“:

```
java -cp classes de.unimarburg.einfachGenial.MainKI Gruppe3 JOIN StupidGame
```

Wenn ihr eine lauffähige JAR-Datei erzeugt, dann wird die Kommando-Zeile wie:

```
java -jar EinfachGenial.jar Gruppe3 JOIN StupidGame
```

Ihr sollt in einer Datei KI-Beschreibung.txt beschreiben, wie man eure KI laufen lassen kann.

3 Allgemeines über Netzwerkunterstützung und KI

Hier fasse ich einige wichtige Anforderungen für Netzwerkunterstützung und KI zusammen, die fürs Bestehen der Prüfung wichtig sind:

- Die KI muss über Netzwerk ein vollständiges Spiel machen können, ohne unmögliche Züge zu machen.
- Eure Client muss beide Server unterstützen, den Server der am Port 34567 läuft , und den Server der am Port 34568 läuft , wie im Netzwerk-Protokoll beschrieben.
- Die KI muss jeden einzelnen Zug innerhalb einer Minute senden.
- Die KI muss gegen „StupidGame“ gewinnen.