

Prof. Dr. Ir. J.-P. Katoen

Viet Yen Nguyen, David R. Piegdon

## Softwarepraktikum SS 2010

### Implementierung heuristischer Algorithmen für Brettspiele

#### – Übungsblatt 1 –

**Abgabe/Besprechung der Lösungen am 5.5.2010 im Seminarraum des Lehrstuhls**

#### **Aufgabe 1**

Entwerfen Sie drei verschiedene Spielfelder sinnvoller Größe gemäß der Spezifikation des Dateiformats und senden Sie diese im ASCII Format bis spätestens Freitag 30.04.2010 per E-Mail an:

`swp10@p23q.org`

#### **Aufgabe 2**

Entwickeln und implementieren Sie eine Datenstruktur zur Repräsentation eines Spielfeldes im Speicher. Bauen Sie Ihre Implementierung modular aber effizient auf. Schreiben Sie außerdem eine Routine, die das Einlesen von beliebigen, spezifikationskonformen Spielfeldern aus Dateien und eine Konvertierung in die von Ihnen entwickelte Datenstruktur aus Aufgabe 2 ermöglicht.

#### **Aufgabe 3**

- Entwerfen Sie einen Algorithmus, der bestimmt, ob ein Zug gültig ist. Berücksichtigen Sie dabei evtl. vorhandene Transitionen, Überschreib- und Expansionssteine (Bomben und Spezialfelder brauchen Sie noch nicht zu berücksichtigen).
- Implementieren Sie diesen Algorithmus aufbauend auf Ihrer in Aufgabe 2 erstellten Datenstruktur zur Spielfeldrepräsentation. Das Programm soll dabei bei Eingabe eines beliebigen Zuges entweder den Nachfolgezustand berechnen (und ausgeben können) oder einen ungültigen Zug erkennen.

#### **Aufgabe 4**

Geben Sie zwei verschiedene *nichttriviale*<sup>1</sup> sinnvolle Funktionen zur Bewertung von beliebigen Spielzuständen an. Der Funktionswert soll dabei die Güte eines Zustandes für einen gegebenen Spieler repräsentieren.

Dokumentieren Sie Ihre Heuristiken ausführlich im Praktikumsbericht.

<sup>1</sup>Dies schließt ein einfaches Abzählen der Spielsteine aus.

## Aufgabe 5

Erweitern Sie Ihr Programm dahingehend, dass Spezialfelder (Wahl-, Inversions-, Bonus- und Expansionsfelder) sowie Überschreibsteine bei der Spielzugberechnung korrekt behandelt werden. Bei Wahlfeldern ist dabei ein zusätzlicher Parameter für die gewählte Farbe und bei Bonusfeldern ein weiterer Parameter für den gewählten Zusatzstein vonnöten.

Testen Sie Ihr Programm anhand von Beispielen, die solche Felder beinhalten.

### Hinweis:

Bitte laden Sie jeglichen von Ihnen geschriebenen Code sowie den Praktikumsbericht (als L<sup>A</sup>T<sub>E</sub>X-Code) ins SVN Repository hoch. Das Repository ist für die Gruppe *X* unter folgender Adresse verfügbar:

<https://svn-i2.informatik.rwth-aachen.de/repos/swpss10/gruppeX>

Der Benutzername ist jeweils **gruppeX**.

Zum jeweiligen Abgabetermin müssen die Lösungen im SVN vorliegen. Sie werden dazu von uns die entsprechenden Accounts und Passwörter zugewiesen bekommen. Sollten Sie mehrfach Lösungen unpünktlich oder fehlerhaft abgeben, so müssen Sie mit dem Ausschluß vom Softwarepraktikum rechnen. Es wird auch feste Deadlines geben, an denen Ihr Programm bestimmte Kriterien wie z.B. Netzwerktauglichkeit erfüllen muss.

Bei weiteren Fragen oder Problemen können Sie den noch von uns einzurichtenden E-Mail-Verteiler nutzen oder sich auch gerne an uns wenden.