

Prof. Dr. Ir. J.-P. Katoen
Viet Yen Nguyen, Christian Dehnert, Gereon Kremer

Softwarepraktikum SS 2012

Implementierung heuristischer Algorithmen für Brettspiele

– Übungsblatt 1 –

Abgabe/Besprechung der Lösungen am 19.04.2012 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, 13.04.2012 per E-Mail an:

`gereon@i2.informatik.rwth-aachen.de`

Aufgabe 2

Entwickeln und implementieren Sie eine Datenstruktur zur Repräsentation eines Spielfeldes im Speicher. Bauen Sie Ihre Implementierung modular aber effizient auf. Die maximale Spielfeldgröße beträgt 100 mal 100 Felder. Schreiben Sie außerdem eine Routine, die das Einlesen von beliebigen, spezifikationskonformen Spielfeldern aus Dateien und eine Konvertierung in die von Ihnen entwickelte Datenstruktur 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*¹ 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.

¹Dies 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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Code) ins GIT Repository hoch. Es existiert ein Repository, das eine $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Vorlage und weitere Dokumentation beinhaltet. Es ist unter folgender Adresse erreichbar:

<https://sselab.de/lab9/private/git/swp-i2-2012-pm>

Das Repository ist für die Gruppe X unter folgender Adresse verfügbar:

<https://sselab.de/lab9/private/git/swp-i2-2012-gX>

Zum jeweiligen Abgabetermin müssen die Lösungen im GIT vorliegen. Sie müssen dazu SSELab-account benutzen. 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 kann und soll der Mailverteiler `swp-i2-2012-pm@lab9.sselab.de` verwendet werden, auf dem alle Teilnehmer und Betreuer eingetragen sind. Alle Teilnehmer sollten nach Möglichkeit auf die Fragen anderer antworten.