Fortgeschrittene Funktionale Programmiernug in Haskell

Projekt: Spielentwicklung in Haskell (Sokoban)

Tutoren: Jonas Betzendahl (`jbetzend@techfak...`), Stefan Dresselhaus (`sdressel@techfak...`)

Aufgabenstellung:

Bei diesem Projekt sollen Sie ein kleines Spiel mit grafischer Oberfläche (GUI) in Haskell schreiben. In Ihrer Implementation soll das Spiel Sokoban spielbar sein, in dem es darum geht, alle Kisten auf markierte Zielpunkte zu schieben (mit nur wenig Platz für Manöver).

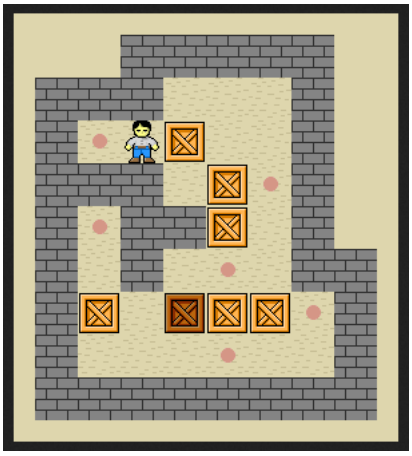

Abbildung 1: Ein mögliches Sokoban-Level. Credit: *Carloseow (Wikipedia)*

Mindestanforderungen:

Das Spiel muss fehlerfrei spielbar sein (also: Siegbedingungen überprüfen, korrekte Kollisionsabfrage, ...), bei erfolgreich absolviertem Level automatisch das nächste Laden und einen Counter enthalten, wie viele Züge in diesem Level bereits getätigt wurden.

Es sollen mindestens zehn spielbare Level vorliegen, die bei Spielstart aus einer oder mehreren externen Textdateien geladen werden sollen. Diese Dateien sollen auch mit einem Texteditor veränderbar sein, um zum Beispiel zusätzliche Level hinzuzufügen.

Eine grafische Oberfläche ist Pflicht (Ausgabe auf der Konsole genügt *nicht*!), die Wahl der GUI-Bibliothek (`sdl2`, `gtk2hs`, ...) steht Ihnen jedoch frei.

Zusatz:

Bei besonderer Motivation können außerdem die folgenden Features noch eingebaut werden:

- **Rückgängig-Funktion:** Ermöglichen Sie es dem Benutzer, seine Züge einen nach dem anderen (bis zum Start des Levels) via Knopf- oder Tastendruck rückgängig zu machen.
- **KI zum Lösen:** Gegeben ein beliebiges Level, implementieren Sie eine Funktion, die ermittelt, ob es für dieses Level eine Lösung gibt und welches die kürzeste Abfolge von Zügen ist, die das Level absolviert.
- **Levelgenerierung:** Anstelle eine bestimmte Anzahl vorgefertigter Level bereit zu stellen, schreiben Sie einen Algorithmus, der ein zufälliges (lösbares, siehe oben) Level generiert. So kann ein Spielmodus erstellt werden, den man theoretisch ewig weiter spielen könnte.

Abgabemodalitäten:

Eine gültige Abgabe ist ein `cabal`-Projekt, das fehlerfrei in einer Sandbox installiert werden kann und eine funktionierende ausführbare Datei generiert (Testumgebung ist im Zweifelsfall wie immer das GZI). Bitte reichen Sie Ihre Projekte spätestens bis zum **Freitag, den 18.09.2015** ein. Dazu schicken Sie alle Dateien, die zu Ihrem Projekt gehören (eventuell modulo einer vernünftigen `.gitignore`) in einem Dateiarchiv an beide Tutoren.

Falls gewünscht, kann Ihnen für die Entwicklung des Projekts ein privates Repository auf `GitHub` zur Verfügung gestellt werden. Dann kann auch direkt dort abgegeben werden. Kontaktieren Sie dafür bitte die Tutoren.

Sollten Sie Rückfragen haben oder Hilfestellung benötigen, wenden Sie sich bitte ebenfalls an die Tutoren.