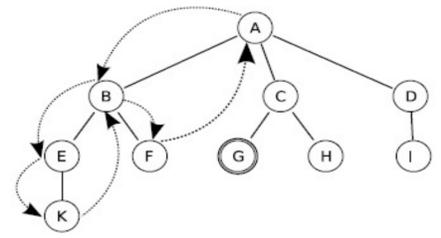
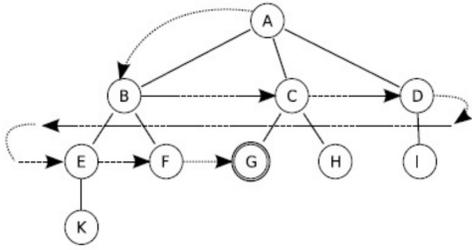


### Entwurf und Implementierung von Suchverfahren am Beispiel des Spiels "Rush-Hour" mit Visualisierung der Lösungsfindung

Bachelorarbeit, vorgelegt von Johann – Sebastian Pleban



#### Aufgabenstellung:

Ziel dieser Arbeit ist es ein Programm zu entwickeln, mit dem sich Suchverfahren Visualisieren lassen.

Dazu werden zunächst die Grundlagen vermittelt. Die Entwicklung des Programms wird über die Entwurfsphase und die Implementierung begleitet.

Hierbei wird Wert auf die Wiederverwendbarkeit der Suchalgorithmen gelegt, so dass sie für andere Suchprobleme verwendbar sind.



Lösung der Beispielwelt<sup>1</sup>

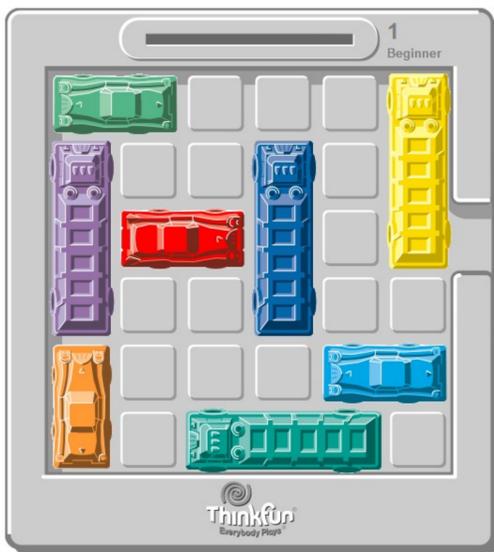
#### Piccolo2D:

Zur Visualisierung der Graphen wurde das Plugin Piccolo2D verwendet. Es baut auf der Grafik-API auf und übernimmt dabei Aufgaben wie automatisches repaint der Oberfläche, die Verwaltung von Events, Animation und Layout.

Piccolo2D ist momentan in drei Versionen, für Java, .NET, und .NET Compact Framework verfügbar.

Piccolo2D ist dabei freie Software (BSD-Lizenz) und Open Source.

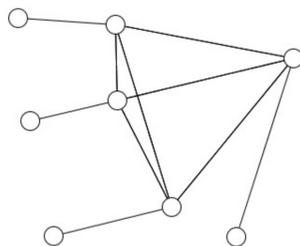
#### Das Spiel "Rush-Hour":



Beispielwelt eines Rush-Hour Spiels<sup>1</sup>

#### Suchalgorithmen:

Es kommen dabei informierte und uninformierte Suchverfahren in Frage. Zu den uninformierten gehören zum Beispiel Tiefensuche und Breitensuche. Als informierter Suchalgorithmus könnte der A\*-Suchalgorithmus zum Einsatz kommen.



Beispielgraph bei einem Zielauto und einem Objekt

#### Programmablauf:

Der Benutzer gibt durch Eingabe eines Strings die Spielwelt vor. Nach Auswahl eines Suchverfahrens beginnt das Programm mit Hilfe der ausgewählten Suche die Spielwelt zu lösen. Dazu werden zu jedem Zustand die möglichen Folgeknoten berechnet, der aktuelle Zustand also expandiert. Die Folgeknoten werden dann entsprechend dem Suchverfahren weiter behandelt.

#### Wiederverwendbarkeit:

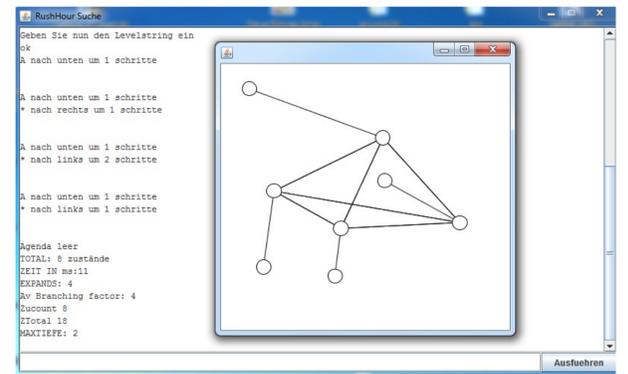
Durch die Trennung von Suchverfahren und dem eigentlichen Problem, ist es möglich, das Programm für andere Suchprobleme zu nutzen.

Die Suchverfahren verlaufen nach immer gleichem Muster, jedoch sind einige Eigenschaften stets problemabhängig.

Dazu gehören:

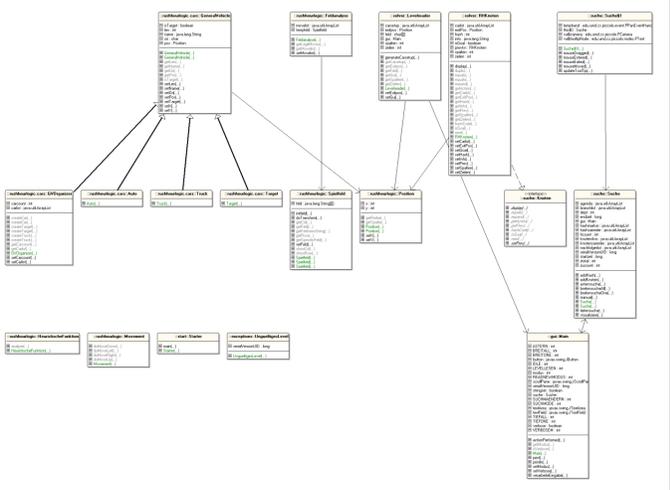
- Ziel erreicht? - Funktion
- Übergangsfunktion (Expandierung)
- Gleichheit der Zustände
- Visualisierung

#### Visualisierung des Suchraums:



Zur Visualisierung der Daten der Algorithmen dient ein Graphical User Interface (GUI). Dies ist angebunden an das Plugin Piccolo2D. Dadurch entstehen zwei Fenster, eines zur Steuerung der Suche, das andere visualisiert den Suchraum.

Das Programm selbst ist dabei textgesteuert. Im Suchraumfenster lassen sich die Knoten frei bewegen. Entwickelt wurde das Programm mit der Eclipse IDE (Version: Juno) und der Java JDK-Version 1.7.



Übersicht über die Programmarchitektur in UML

<sup>1</sup>: Programmiert von Mark Riedel, Bilder entnommen von <http://www.puzzles.com/products/RushHour/RHfromMarkRiedel/Jam.html>  
Rush-Hour - Copyright © 2003-2012 ThinkFun Inc.