



Pause  
Neustart  
Beenden

Fps: 63

Bewegungen: 22

Drehungen: 9

Gesamt: 32

Geschwindigkeit: 15

Wiederholung: Nein

Spiefeld: chop.dat

Karten: <Zufall>

Schwierigkeit: schwer

Status: Gewonnen

DER WEG IST DAS ZIEL | Ein schlauer Roboter sucht den direkten Weg zum Ziel.

## Visualisierungshilfe: Die GUI zum Programmierwettbewerb

Von: Dominik Honnef/  
Dominik Wagenführ

Da beim letzten Programmierwettbewerb einige Nutzer nach einer grafischen Oberfläche (GUI) fragten, können wir hier stolz eine präsentieren.

**B**eim letzten Programmierwettbewerb von freiesMagazin im April dieses Jahres [1] fragten einige Nutzer nach einer grafischen Oberfläche (GUI), um das Geschehen besser verfolgen zu können. Aus diesem Grund hat Dominik Honnef diese für den anstehenden Wettbewerb umgesetzt. Die GUI ist dabei als Programmier- und Visualisierungshilfe gedacht und wird nicht zwingend für eine Teilnahme benötigt.

### ANFORDERUNGEN

Für die Umsetzung wird Ruby [2], Ruby-Gnome2 [3], Ruby-Gems [4] und die Gosu Game Development Library [5] benutzt. Daher muss man ein paar

Ruby-Pakete (und deren Abhängigkeiten) installieren:

- **ruby** bzw. **ruby1.8**
- **rubygems**
- **libgtk2-ruby** bzw. **libgtk2-ruby1.8**

Daneben benötigt man noch einige Entwickler-Bibliotheken, um die Gosu GUI kompilieren zu können:

- **g++**
- **libgl1-mesa-dev**
- **libpango1.0-dev**
- **libboost-dev**
- **libsdl-mixer1.2-dev**
- **ruby1.8-dev**

Die Paketbezeichnung stammt von Debian/Ubuntu und

kann/wird unter anderen Distributionen variieren. Unter Arch Linux gibt es beispielsweise keine extra dev-Pakete. Hier hilft ein Blick in die offizielle Installationsanleitung [6] weiter.

Nach der Installation der Pakete installiert man die Gosu GUI Library mit Root-Rechten über

```
# gem install gosu
```

Falls gefragt wird, welches gosu-Paket installiert werden soll, wählt man am besten **gosu 0.7.14 (ruby)** aus.

Falls eine Fehlermeldung der Art

```
ERROR: While \
executing gem ... (\
Gem::\
GemNotFoundException)
```