

# L<sup>A</sup>T<sub>E</sub>X-Workshop

## Magazinerstellung mit L<sup>A</sup>T<sub>E</sub>X

Dominik Wagenführ

Ubucon 2009, Universität Göttingen

17. Oktober 2009

- 1 Einleitung
- 2 Textformatierung
- 3 Layout
- 4 Bilder
- 5 Optionale Parameter

- Wissenschaftliche Arbeiten mit L<sup>A</sup>T<sub>E</sub>X bekannt
- Aber auch Magazingestaltung möglich
- Probleme dabei: Bilder, Bilder, Bilder
  - hauptsächlich wegen absoluter Positionierung ...  
... und textumflossenen Bereichen
- „Normaler“ Textsatz aber auch anspruchsvoll



- Rohtext in Datei `artikel0-rohtext.tex`
- Präambel:

Beispiel `artikel1-rohform.tex`

```
\documentclass[twoside]{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{ngerman}
\usepackage[expansion=false]{microtype}
\usepackage[scaled]{helvet}

\renewcommand*{\familydefault}{\sfdefault}
\renewcommand*{\ttdefault}{\ttt}
```

- Dokumentklasse KOMA-Script<sup>[1]</sup> `scrartcl` mit zweiseitigem Layout `twoside`
- UTF8-Kodierung `utf8`, u. a. für Darstellung deutscher Umlaute
- T1-Schriftkodierung<sup>[2]</sup>
- Neue Deutsche Rechtschreibung mit Paket `ngerman` (u. a. für Trennungen)
- Paket `microtype`<sup>[3]</sup> für besseres Schriftbild, Option `expansion=false` ohne „Font Expansion“
- Serifenlose Schrift `helvet` (Helvetica) in skalierter Form (72 %)
- Serifenlose Schrift als Standard: `\sfdefault`
- Für dicktengleiche Schrift: `txtt`

---

[1] <http://www.komascript.de/>

[2] <http://de.wikipedia.org/wiki/T1-Kodierung>

[3] <http://www.ctan.org/tex-archive/help/Catalogue/entries/microtype.html>

## Weitere Vorabformatierungen:

- Überschriften mit `\section`
- Listen mit `itemize`
- Befehle mit `verbatim`
- Quellen mit `enumerate`
- Sonderzeichen wie „, “ und &



## Menüs

Beispiel artikel2-format1.tex

```
\newcommand*{\Menu}[1]{\textit{"'#1'}}
```

- „*Menü-Einträge*“ sollen kursiv mit Anführungszeichen werden
- Dazu zählen auch Schaltflächen, Checkboxen etc.
- Anwendung:

```
Über \Menu{Neues Spiel starten} kann man einstellen,  
...
```

## 1.2 GUI-Start

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche der Einstellungen des Spiels „robots-gui-helper“ und zum anderen aus der Oberfläche des Spielbretts „robots-gui“. Da die „robots-gui“ nie direkt gestartet werden sollte, werden die Optionen hier nicht näher erläutert.

Den GUI-Starter ruft man per

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „*Spiel starten*“ klicken und die Roboter sollte den Roboter bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die Einstellungen eingegangen werden, die nicht offensichtlich sind.

Über „*Neues Spiel starten*“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei „bot.txt“ oder von der Startposition auf dem Spielbrett starten soll. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn der Haufen gesetzt ist. Auf die Art kann man ein unterbrochenes Spiel fortsetzen.

Entfernt man den Haken bei „*Spiel animieren*“, läuft der Roboter nicht über das Spielfeld, sondern das Spiel wird bis zum Ende berechnet, der Roboter also das Ziel gefunden oder zerstört wurde. Erst dann wird die Anzeige freigeschaltet und man sieht den Roboter an der Endposition.

*Abbildung: artikel2-format1.pdf*

## Pakete

Beispiel artikel2-format2.tex

```
\newcommand*{\Paket}[1]{\textbf{\mbox{#1}}}
```

- **Pakete** werden fett markiert
- Anwendung:

```
\begin{itemize}  
  \item \Paket{ruby} bzw. \Paket{ruby1.8}  
  \item ...  
\end{itemize}
```

nef diese für den anstehenden Wettbewerb umgesetzt. Die GUI ist dabei als Programmier- und Visualisierungshilfe gedacht und wird nicht zwingend für eine Teilnahme benötigt.

## 1.1 Anforderungen

Für die Umsetzung wird Ruby [<http://www.ruby-lang.org/de/>], Ruby-Gnome2 [<http://ruby-gnome2.org/>], Ruby-Gems [<http://gems.rubyforge.org/>] und die Gosu Game Development Library [<http://www.libgdx.com/>] 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**

*Abbildung: artikel2-format2.pdf*

## Terminaleingaben

Beispiel `artikel2-format3.tex`

```
\newcommand*{\term}[1]{\texttt{\textbf{#1}}}
```

- Terminaleingaben im Fließtext mit dicktengleicher Schrift (Monotype/Typewriter)
- Dazu zählen auch Dateien und Ordner!
- Anwendung:

```
Das Standardthema im Ordner \term{images/new} wurde  
...
```

## 1.2 GUI-Start

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche der Einstellungen des Spiels **robots-gui-helper** und zum anderen aus der Oberfläche des Spielbretts **robots-gui**. Da die **robots-gui** nie direkt gestartet werden sollte dessen Optionen hier nicht näher erläutert.

Den GUI-Start ruft man per

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „*Spiel starten*“ klicken und die Roboter sollte den Roboter bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die eingegangen werden, die nicht offensichtlich sind.

Über „*Neues Spiel starten*“ kann man einstellen, ob der Roboter von seiner le position aus der Datei **bot.txt** oder von der Startposition auf dem Spielbrett startet. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn das nicht gesetzt ist. Auf die Art kann man ein unterbrochenes Spiel fortsetzen.

Entfernt man den Haken bei „*Spiel animieren*“, läuft der Roboter nicht über das sondern das Spiel wird bis zum Ende berechnet, der Roboter also das Ziel gefunden oder zerstört wurde. Erst dann wird die Anzeige freigeschaltet und man sieht den Roboter an der Endposition.

*Abbildung: artikel2-format3.pdf*

## Tasteneingaben

Beispiel artikel2-format4.tex

```
\usepackage [ngerman] {keystroke}
\newcommand*{\Taste}[1]{{\small\keystroke{#1}}}
```

- Paket `keystroke`<sup>[4]</sup> für Tasten
- Option `ngerman` für deutsche Bezeichnung:  
 ,  ,  etc.
- Anwendung:

Mit den Tasten `\Taste{+}` und `\Taste{-}` ...

<sup>[4]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/keystroke.html>

# Einfache Formatierungen

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

Über „*Neues Spiel starten*“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei **bot.txt** oder von der Startposition auf dem Spielbrett starten soll. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn der Haken nicht gesetzt ist. Auf die Art kann man ein unterbrochenes Spiel fortsetzen.

Entfernt man den Haken bei „*Spiel animieren*“, läuft der Roboter nicht über das Spielfeld, sondern das Spiel wird bis zum Ende berechnet, der Roboter also das Ziel gefunden hat oder zerstört wurde. Erst dann wird die Anzeige freigeschaltet und man sieht den Roboter an der Endposition.

Mit „*Spiel starten*“ startet man das Spiel und kann dem Roboter bei der Bewegung zuschauen. Der Roboter hinterlässt eine transparente grüne Spur (Nein, er verliert kein Öl!), mit der man nachvollziehen kann, welche Felder er wie häufig betreten hat. Mit den Tasten  und  kann man die Bewegungsgeschwindigkeit des Roboters verändern.

Die „grüne Spur“, d. h. die Bewegungen des Roboters werden während des Ablaufes in der Sequenzdatei abgespeichert, die im zugehörigen Feld angegeben ist (als Standard ist die Datei **globalseq.txt** eingetragen). Diese Datei hat beispielsweise folgenden Aufbau, wobei jede Zeile einer Bewegung oder Drehung entspricht:

MU

RL

MR

----

2

Abbildung: *artikel2-format4.pdf*

## Beispiel artikel3-ueberschrift1.tex

```
\usepackage{titlesec}
\titlespacing{\section}{0mm}{2em}{1em}
\titlespacing{\subsection}{0em}{2.5mm}{0mm}
```

- Paket `titlesec`<sup>[5]</sup> für leichte Redefinition von Überschriften
- `\titlespacing` für Abstände, Argumente:
  - 1 Überschriftenebene
  - 2 Linke Einrückung
  - 3 Abstand vor Überschrift
  - 4 Abstand nach Überschrift

---

<sup>[5]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/titlesec.html>

## Beispiel artikel3-ueberschrift1.tex

```
\usepackage{titlesec}
\titleformat{\section}{\normalfont\Huge}{}{0mm}{}
\titleformat{\subsection}%
{\normalfont\large\bfseries\scshape}{}{0mm}{}

```

- `\titleformat` für Gestaltung, Argumente:
  - 1 Überschriftenebene
  - 2 Format
  - 3 Nummerierung (Label)
  - 4 Abstand zwischen Label und Überschrift
  - 5 Auszuführender Code vor Überschrift

## 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.

Beim letzten Programmierwettbewerb von freiesMagazin im April dieses Jahres [<http://www.freiesmagazin.de/20090> programmierwettbewerb-gestartet] 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 [<http://www.ruby-lang.org/de/>], Ruby-Gnome2 [<http://ruby-gnome2.sourceforge.jp/de/>], Ruby-Gems [<http://gems.rubyforge.org/>] und die Gosu Game Development Library [<http://www.libgosu.org/>] 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:

*Abbildung: [artikel3-ueberschrift1.pdf](#)*

## Beispiel artikel3-ueberschrift2.tex

```
\usepackage{color}
\definecolor{mittelblau}{rgb}{0.1,0.5,1.0}
\newcommand{\colbox}[3]{\colorbox{#1}{\textcolor{#2}{#3}}}
```

- Paket `color` für Farben (68 sind vordefiniert)<sup>[6]</sup>
- `\definecolor` für Definition neuer Farben, Argumente:
  - 1 Farbname
  - 2 Farbmodus: `rgb`, `gray`, `cmYk`
  - 3 Farbdefinition (bei RGB drei, bei Grau ein, bei CMYK vier Werte)
- Definition neuer Box `\colbox`:
  - `\textcolor` setzt Farbe #2 für Text #3
  - `\colorbox` füllt Box mit einer Farbe #1

---

<sup>[6]</sup><http://en.wikibooks.org/wiki/LaTeX/Colors>

## Beispiel artikel3-ueberschrift2.tex

```
\newcommand*\maincolor{mittelblau}
\newcommand*\maintextcolor{white}
\newcommand*\SetMainColor[1]
  {\renewcommand*\maincolor{#1}}
\newcommand*\SetMainTextColor[1]
  {\renewcommand*\maintextcolor{#1}}
```

- Definition einer Hauptfarbe als Thema
- Definition einer Haupttextfarbe in Kontrast zur Hauptfarbe
- Definitionen für Änderung der Farben

## Beispiel artikel3-ueberschrift2.tex

```
\usepackage{calc}
\newcommand{\Abschnitt}[1]{%
  \setlength{\fboxsep}{0.2em}%
  \subsection*{\colbox{\maincolor}{\maintextcolor}%
    {\parbox{\linewidth-2\fboxsep-2\fboxrule}{#1}}}%
  \noindent{}}
}
```

- Paket `calc` für einfache Berechnungen (hilfreich bei Längen)
- Definition des Abschnittes:
  - Länge `\fboxsep` definiert Abstand zwischen Rand und Text in einer Box
  - `\parbox` für eine Box mit spezifischer Breite (und optional Höhe)
  - `\linewidth-2\fboxsep-2\fboxrule` füllt genau die Seitenbreite
  - `\noindent`, damit nachfolgender Text nicht eingerückt wird

## 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.

Beim letzten Programmierwettbewerb von freiesMagazin im April dieses Jahres [<http://www.freiesmagazin.de/20090>] programmierwettbewerb-gestartet) 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 [<http://www.ruby-lang.org/de/>], Ruby-Gnome2 [<http://ruby-gnome2.sourceforge.jp/de/>], Ruby-Gems [<http://gems.rubyforge.org/>] und die Gosu Game Development Library [<http://www.libgosu.org/>] 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++**

*Abbildung: artikel3-ueberschrift2.pdf*

# Überschriften

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

## Beispiel artikel3-ueberschrift3.tex

```
\usepackage{lettrine}
\newcommand{\Initial}[1]{%
  \setcounter{DefaultLines}{3}%
  \renewcommand*{\DefaultLoversize}{-0.2}%
  \setlength{\fboxsep}{0.4em}%
  \lettrine[nindent=-0.1mm,findent=0.7em,lraise=0.3]{%
    \colbox{\maincolor}{\maintextcolor}{%
      \begin{minipage}{0.9em}%
        \centering{}%
        #1 %
      \end{minipage}%
    }%
  }{}%
```

## Visualisierungshilfe: Die GUI zum Programmierwettbewerb

von: Dominik Honnel/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 [<http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet>] fragten einige Nutzer nach einer grafischen Oberfläche (GUI), um das Geschehen besser verfolgen zu können. Aus diesem Grund hat Dominik Honnel 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 [<http://www.ruby-lang.org/de/>], Ruby-Gnome2 [<http://ruby-gnome2.sourceforge.jp/de/>], Ruby-Gems [<http://gems.rubyforge.org/>] und die Gosu Game Development Library [<http://www.libgosu.org/>] 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:

*Abbildung: artikel3-ueberschrift3.pdf*

# Überschriften

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

- Paket `lettrine`<sup>[7]</sup> für große Anfangsbuchstaben (Initialen)
- Zähler `DefaultLines` gibt Standardhöhe für die Initiale an (hier: 3)
- `\DefaultLoversize` verkleinert die Initiale
- `nindent` ist die Einrückung für alle Zeilen außer der ersten
- `findent` ist die Einrückung für alle Zeilen
- `lraise` „hebt“ die Initiale in der Zeile etwas an
- Zum Schluss Definition einer Box mit der Initiale
- Anwendung:

```
\Initial{B}eim letzten Programmierwettbewerb ...  
...  
\Abschnitt{Anforderungen}%  
Für die Umsetzung wird Ruby ...
```

<sup>[7]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/lettrine.html>

## Beispiel artikel4-listen1.tex

```
\usepackage{amssymb}
\renewcommand*{\labelitemi}{%
  \textcolor{\maincolor}{\footnotesize$\blacksquare$}%
}
```

- Redefinition der Auflistungszeichen über `\labelitemi`
- Paket `amssymb` für mathematische Symbole, hier für ■

## ANFORDERUNGEN

Für die Umsetzung wird Ruby [<http://www.ruby-lang.org/de/>], Ruby-Gnome2 [<http://ruby-gnome2.sourceforge.jp/de/>], Ruby-Gems [<http://gems.rubyforge.org/>] und die Gosu Game Development Library [<http://www.libgosu.org/>] 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++**
- **libgl-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 [<http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide>] weiter.

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

*Abbildung: artikel4-listen1.pdf*

## Beispiel artikel4-listen2.tex

```
\usepackage{enumitem}
\newenvironment{Auflistung}
{\begin{itemize}[itemsep=0.0em,leftmargin=*]}
{\end{itemize}}
```

- Paket `enumitem`<sup>[8]</sup> für Auflistungen ohne Einrückung
- Neue Umgebung für Auflistungen:
  - `itemsep` – Abstand zwischen Zeilen in einem Eintrag
  - `leftmargin=*` – Verhindert Einrückung
  - Anwendung:

```
\begin{Auflistung}
\item ...
\end{Auflistung}
```

<sup>[8]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/enumitem.html>

## ANFORDERUNGEN

Für die Umsetzung wird Ruby [<http://www.ruby-lang.org/de/>], Ruby-Gnome2 [<http://ruby-gnome2.sourceforge.jp/de/>], Ruby-Gems [<http://gems.rubyforge.org/>] und die Gosu Game Development Library [<http://www.libgosu.org/>] 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 [<http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide>] 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.

*Abbildung: artikel4-listen2.pdf*

## Beispiel artikel4-listen3.tex

```
\usepackage{enumitem}
\newenvironment{Aufzaehlung}
{\begin{enumerate}[itemsep=0.0em,leftmargin=*]}
{\end{enumerate}}
```

- Analog für Aufzählungen (nummerierte Listen)
- Anwendung:

```
\begin{Aufzaehlung}
\item ...
\end{Aufzaehlung}
```

beeinflussen.

Das Standardthema im Ordner **images/new** wurde von Arne Weinberg erstellt. Die Bilder unterliegen der GFDL [<http://www.gnu.org/licenses/fdl.html>].

Alternativ gibt es noch eine handgemalte Version im Ordner **images/pen**. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative-Commons-Lizenz „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“ [<http://creativecommons.org/licenses/by-sa/3.0/de/>].

Autoreninfo: Dominik Honnef und Dominik Wagenführ  
Beide Autoren sind Redakteure bei freiesMagazin und auch Programmierer aus Leidenschaft.

Verweise:

1. freiesMagazin-Wettbewerb: <http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet>
2. Ruby: <http://www.ruby-lang.org/de/>
3. Ruby-Gnome2: <http://ruby-gnome2.sourceforge.jp/de/>
4. RubyGems: <http://gems.rubyforge.org/>
5. Gosu Game Development Library: <http://www.libgosu.org/>
6. Ruby-Gnome2 Installation: <http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide>
7. GFDL: <http://www.gnu.org/licenses/fdl.html>
8. CC-BY-SA-3.0: <http://creativecommons.org/licenses/by-sa/3.0/de/>

*Abbildung: artikel4-listen3.pdf*

# Links und Quellen

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

## Beispiel artikel5-links1.tex

```
\usepackage[
  bookmarks=true,
  bookmarksopen=true,
  colorlinks=true,
  urlcolor=url,
  pdfauthor={Dominik Wagenführ},
  pdfcreator={Dominik Wagenführ},
  pdfsubject={Supertolles Magazin!},
  pdfkeywords={Linux, Open Source}
]{hyperref}
```

- Paket `hyperref`<sup>[9]</sup> für Links
- Paketoptionen:
  - `bookmarks` – Inhaltsverzeichnis in PDF einbetten
  - `bookmarksopen` – Inhaltsverzeichnis automatisch aufklappen
  - `colorlinks` – färbt Linktext ein, dafür kein Rahmen
  - `urlcolor` – Farbe für Linktext
  - `pdfauthor` – Autor des Dokuments
  - `pdfcreator` – Ersteller des Dokuments
  - `pdfsubject` – Beschreibung des Dokuments
  - `pdfkeywords` – kommaseparierte Stichworte zum Dokument

---

<sup>[9]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/hyperref.html>

## Beispiel artikel5-links1.tex

```
\definecolor{url}{rgb}{0,0,0}
\newcounter{linkcounter}
\newcommand*{\Link}[2]%
  [\arabic{linkcounter}\stepcounter{linkcounter}]%
  {\~\href{#2}{[#1]}}
```

- Definition der Linkfarbe `url` (hier: schwarz)
- Neuer Zähler `linkcounter` für Links
- Neuer Befehl `\Link`:
  - Erhöht per Standard den Zähler um 1
  - `\href` verlinkt (mit #2) einen Text ([#1])
  - Steht durch die Tilde nicht einzeln auf einer Zeile

- Normale Anwendung:

```
Für die Umsetzung wird  
Ruby\Link{http://www.ruby-lang.org/de/} ...
```

- Sonderfall doppelter Link:

```
Wie bereits erwähnt braucht man  
Ruby\Link[2]{http://www.ruby-lang.org/de/} ...
```

## ANFORDERUNGEN

Für die Umsetzung wird Ruby [2], Ruby-Gnome2 [3], Ruby-Gems [4] und die Gosu 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**

*Abbildung: artikel5-links1.pdf*

# Links und Quellen

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

## Beispiel artikel5-links2.tex

```
\newenvironment{Quellen}[1][\linewidth]
{%
  \noindent{}%
  \begin{minipage}{#1}
    \hfill{}%
    \begin{tabular}{r}
      \textcolor{dunkelgrau}%
        {\normalfont\large\scshape Verweise}\ll[-1.9ex]
      \textcolor{\maincolor}{\rule{2cm}{1.5pt}}\ll[-0.8ex]
    \end{tabular}
    \renewcommand*{\labelenumi}{\theenumi}
    \begin{small}
      \begin{Aufzaehlung}
}
{%
      \end{Aufzaehlung}
    \end{small}
    \renewcommand*{\labelenumi}{\theenumi.}
  \end{minipage}
}
```

- Neue Umgebung für Quellen
- Breite gesteuert über Minipage
- Überschrift und Linie eingebettet in rechtsbündige Tabelle
- Aufzählung wie Links im Text: [X]
- Anwendung: Ersetze Aufzaehlung durch Quellen:

```
\begin{Quellen}
\item freiesMagazin-Wettbewerb: http://www...
\item ...
\end{Quellen}
```

unterliegen der GFDL [7].

Alternativ gibt es noch eine handgemalte Version im Ordner **images/pen**. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative-Commons-Lizenz „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“ [8].

Autoreninfo: Dominik Honnef und Dominik Wagenführ

Beide Autoren sind Redakteure bei **freiesMagazin** und auch Programmierer aus Leidenschaft.

## VERWEISE

- [1] freiesMagazin-Wettbewerb: <http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet>
- [2] Ruby: <http://www.ruby-lang.org/de/>
- [3] Ruby-Gnome2: <http://ruby-gnome2.sourceforge.jp/de/>
- [4] RubyGems: <http://gems.rubyforge.org/>
- [5] Gosu Game Development Library: <http://www.libgosu.org/>
- [6] Ruby-Gnome2 Installation: <http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide>
- [7] GFDL: <http://www.gnu.org/licenses/fdl.html>
- [8] CC-BY-SA-3.0: <http://creativecommons.org/licenses/by-sa/3.0/de/>

*Abbildung: artikel5-links2.pdf*

## Beispiel artikel5-links3.tex

```
\newcommand*{\Quelle}[3]{\item #1{}: \href{#2}{#3}}
```

- Neues Kommando für eine Quelle
- Anwendung:

```
\begin{Quellen}  
\Quelle{freiesMagazin-Wettbewerb}{http://www.freies...}%  
    {www.freies...}  
\item ...  
\end{Quellen}
```

- Direkte Verlinkung ohne Argument #3 möglich durch `\url{#2}` in Quelle-Definition

unterliegen der GFDL [7].

Alternativ gibt es noch eine handgemalte Version im Ordner **images/pen**. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative-Commons-Lizenz „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“ [8].

Autoreninfo: Dominik Honnef und Dominik Wagenführ

Beide Autoren sind Redakteure bei **freiesMagazin** und auch Programmierer aus Leidenschaft.

## VERWEISE

- [1] freiesMagazin-Wettbewerb: [www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet](http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet)
- [2] Ruby: [www.ruby-lang.org/de](http://www.ruby-lang.org/de)
- [3] Ruby-Gnome2: [ruby-gnome2.sourceforge.jp/de](http://ruby-gnome2.sourceforge.jp/de)
- [4] RubyGems: [gems.rubyforge.org](http://gems.rubyforge.org)
- [5] Gosu Game Development Library: [www.libgosu.org](http://www.libgosu.org)
- [6] Ruby-Gnome2 Installation: [ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide](http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide)
- [7] GFDL: [www.gnu.org/licenses/fdl.html](http://www.gnu.org/licenses/fdl.html)
- [8] CC-BY-SA-3.0: [creativecommons.org/licenses/by-sa/3.0/de](http://creativecommons.org/licenses/by-sa/3.0/de)

*Abbildung: artikel5-links3.pdf*

# Listings und Befehle

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

## Beispiel artikel6-listings1.tex

```
\usepackage{listings}
\lstdefinestyle{StyleListingBasic}{%
    basicstyle=\ttfamily,
    frame=single,
    framesep=1pt,
    framerule=0.8pt,
    rulecolor=\color{dunkelgrau},
    breaklines=true,
    breakindent=0pt
}
\newcommand*{\term}[1]{%
    \lstinline[style=StyleListingBasic,%
        basicstyle=\ttfamily\bfseries]|#1|
}
```

- Paket `listings`<sup>[10]</sup> für Codelistings und Ähnliches
- Stildefinition über `lstdefinestyle`:
  - `basicstyle` – Schriftstil
  - `frame` – Rahmenart: u. a. `none`, `single`, `shadowbox`
  - `framesep` – Abstand zwischen Rahmen und innerem Text
  - `framerule` – Rahmenbreite
  - `rulecolor` – Rahmenfarbe
  - `breaklines` – zu lange Zeilen werden umgebrochen
  - `breakindent` – Einrückung bei umgebrochenen Zeilen
- Neue Definition für `\term`, da man so auch L<sup>A</sup>T<sub>E</sub>X-Sonderzeichen wie `$`, `-`, `\` etc. nutzen kann.

---

<sup>[10]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/listings.html>

kommt, sollte man die Installation einfach wiederholen, bis das Paket gosu gefunden ist.  
Nach der Kompilierung von Gosu kann man die Entwicklerpakete wieder deinstallieren. Die Pakete von Ruby & Co. müssen natürlich installiert bleiben.

## GUI-START

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche zum Einstellen des Spiels **robots-gui-helper** und zum anderen aus der Oberfläche zum Starten des Spielbretts **robots-gui**. Da die **robots-gui** nie direkt gestartet werden sollte, dessen Optionen hier nicht näher erläutert.

Den GUI-Starter ruft man per

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „*Spiel starten*“ klicken und die Roboter sollte den Roboter bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.

Über „*Neues Spiel starten*“ kann man einstellen, ob der Roboter von seiner Startposition aus der Datei **bot.txt** oder von der Startposition auf dem Spielbrett startet. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn die Datei nicht gesetzt ist. So kann man ein unterbrochenes Spiel fortsetzen.

Entfernt man den Haken bei „*Spiel animieren*“, läuft der Roboter nicht über das

*Abbildung: artikel6-listings1.pdf*

# Listings und Befehle

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

Beispiel artikel6-listings2.tex

```
\definecolor{mittelgrau}{gray}{0.85}
\lstdefinestyle{StyleCommand}{%
    style=StyleListingBasic,
    backgroundcolor=\color{mittelgrau},
    prebreak=\mbox{\textbackslash{}}
}
\lstnewenvironment{Befehl}[1][1]
{\lstset{style=StyleCommand,linewidth=#1\linewidth}}
{}
```

- Listingstil für Befehle im Terminal
- Hintergrundfarbe mittelgrau
- Umbruch wird durch Backslash \ gekennzeichnet
- Neue Listingumgebung `Befehl`, Breite variabel über erstes Argument
- Anwendung:

```
\begin{Befehl}  
$ ./robots-gui-helper  
\end{Befehl}
```

# Listings und Befehle

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

```
Could not find gosu (> 0) in any repository
```

kommt, sollte man die Installation einfach wiederholen, bis das Paket gosu gefunden ist.  
Nach der Kompilierung von Gosu kann man die Entwicklerpakete wieder deinstallieren. Die Pakete von Ruby & Co. müssen natürlich installiert bleiben.

## GUI-START

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche zum Einstellen der Einstellungen des Spiels **robots-gui-helper** und zum anderen aus der Oberfläche des Spielbretts **robots-gui**. Da die **robots-gui** nie direkt gestartet werden sollte, werden dessen Optionen hier nicht näher erläutert.

Den GUI-Starter ruft man per

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „Spiel starten“ klicken und die Roboter sollten sich bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.

Über „Neues Spiel starten“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei **bot.txt** oder von der Startposition auf dem Spielbrett starten soll. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn die Datei nicht gesetzt ist. So kann man ein unterbrochenes Spiel fortsetzen.

Abbildung: *artikel6-listings2.pdf*

# Listings und Befehle

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

## Beispiel artikel6-listings3.tex

```
\definecolor{hellgelb}{rgb}{1.0,1.0,0.9}
\lstdefinestyle{StyleListing}{%
    style=StyleListingBasic,
    backgroundcolor=\color{hellgelb}
}
\lstnewenvironment{Listing}[1][1]
{\lstset{style=StyleListing,linewidth=#1\linewidth}}
{}
```

# Listings und Befehle

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

- Listingstil für Listings im Texteditor
- Hintergrundfarbe hellgelb
- Neue Listingumgebung `Listing`, Breite ebenfalls variabel über erstes Argument
- Anwendung:

```
\begin{Listing}  
MU  
RL  
...  
\end{Listing}
```

# Listings und Befehle

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

an der Endposition.

Mit „*Spiel starten*“ startet man das Spiel und kann dem Roboter bei der Bewegung zuschauen. Der Roboter hinterlässt eine transparente grüne Spur (Nein, er verliert kein Öl!), mit der man nachvollziehen kann, welche Felder er wie häufig betreten hat. Mit den Tasten  und  kann man die Bewegungsgeschwindigkeit des Roboters verändern.

Die „grüne Spur“, d. h. die Bewegungen des Roboters werden während des Ablaufes in der Sequenzdatei abgespeichert, die im zugehörigen Feld angegeben ist (als Standard ist die Datei `globalseq.txt` eingetragen). Diese Datei hat beispielsweise folgenden Aufbau, wobei jede Zeile einer Bewegung oder Drehung entspricht:

```
MU
RL
MR
----
MD
RU
RR
RR
```

2

Abbildung: *artikel6-listings3.pdf*

# Autoreninformation

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung

Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

## Beispiel artikel7-autoreninfo.tex

```
\definecolor{hellgrau}{gray}{0.93}
\newcommand{\Autoreninfo}[3][1.0]{%
  \setlength{\fboxsep}{0pt}%
  \vskip1em}%
  \noindent\colbox{hellgrau}{black}{%
    \parbox{#1\linewidth-2\fboxsep}{%
      \setlength{\fboxsep}{4pt}%
      \colbox{dunkelgrau}{white}{%
        \parbox{\linewidth-2\fboxsep}{{\Large #2}}%
      }\\[0.3em]%
    }\colbox{hellgrau}{black}{%
      \parbox{\linewidth-2\fboxsep}{#3}%
    }\\[0.3em]%
  }%
}
```

# Autoreninformation

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Grobfassung  
Einfachformat

Überschriften

Listen

Links

Listings

Autoreninfo

Layout

Bilder

Parameter

- In der Autoreninfo sollen Autor/en und Text in verschiedenenfarbige Boxen.
- Daher eine große Box für hellgrauen Farbhintergrund.
- Darin eine dunkelgraue Box für Autoren.
- Und darunter eine hellgraue Box mit dem Infotext.
- Anwendung:

```
\Autoreninfo{Dominik Honnef und\\ Dominik Wagenführ}{%  
    Beide Autoren sind Redakteure bei \fm{} und  
    auch Programmierer aus Leidenschaft.%  
}
```

Das Standardbild im Ordner `images/new` wurde von Anne Weimberg erstellt. Die Bilder unterliegen der GFDL [7].

Alternativ gibt es noch eine handgemalte Version im Ordner `images/pen`. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative-Commons-Lizenz „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“ [8].

Dominik Honnef und  
Dominik Wagenführ

Beide Autoren sind Redakteure bei freiesMagazin und auch Programmierer aus Leidenschaft.

## VERWEISE

- [1] freiesMagazin-Wettbewerb: [www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet](http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet)
- [2] Ruby: [www.ruby-lang.org/de](http://www.ruby-lang.org/de)
- [3] Ruby-Gnome2: [ruby-gnome2.sourceforge.jp/de](http://ruby-gnome2.sourceforge.jp/de)
- [4] RubyGems: [gems.rubyforge.org](http://gems.rubyforge.org)
- [5] Gosu Game Development Library: [www.libgosu.org](http://www.libgosu.org)
- [6] Ruby-Gnome2 Installation: [ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide](http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide)
- [7] GFDL: [www.gnu.org/licenses/fdl.html](http://www.gnu.org/licenses/fdl.html)
- [8] CC-BY-SA-3.0: [creativecommons.org/licenses/by-sa/3.0/de](http://creativecommons.org/licenses/by-sa/3.0/de)

*Abbildung: artikel7-autoreninfo.pdf*

## Beispiel artikel8-layout1.tex

```
\documentclass[twoside,9pt]{scrartcl}
```

- Textgröße auf 9pt reicht für Ausdruck
- Für reine Online-Publikation zu klein!

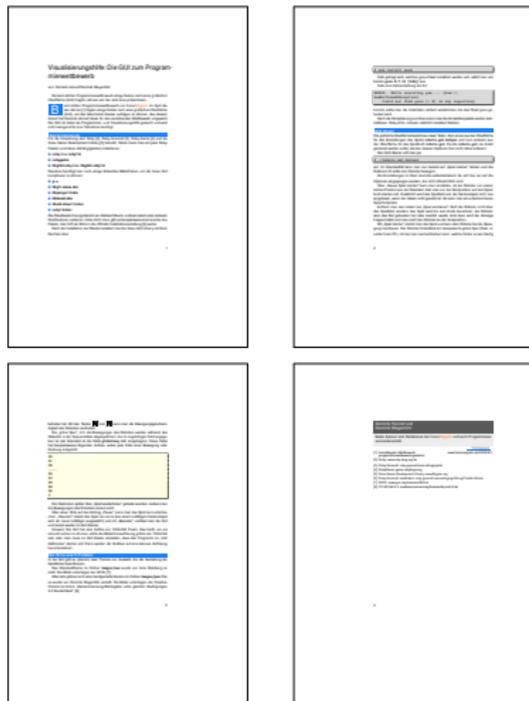


Abbildung: artikel8-layout1.pdf

## Beispiel artikel8-layout2.tex

```
\usepackage{geometry}  
\geometry{text={182mm,261mm}}
```

- Paket `geometry`<sup>[11]</sup> für Satzspiegel
- Festlegung des Textbereiches per `text={BREITE,HOEHE}`
- Ränder sowie Kopf- und Fußleiste passen sich automatisch an
- Weitere Angaben für direkte Randformatierung in `geometry` möglich
- Angaben auch direkt beim Laden des Paketes möglich:

```
\usepackage [text={182mm,261mm}] {geometry}
```

<sup>[11]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/geometry.html>

# Satzspiegel

## Visualisierungshilfe: Die GUI zum Programmierwettbewerb

von Dominik Harnel/Dominik Wagenführ

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

**B**eim letzten Programmierwettbewerb von [heiseopen](#) 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 Harnel diese für die anderen Wettbewerbs umgesehen. Die GUI ist dabei als Programmier- und Visualisierungshilfe gedacht und wird nicht zwingend als Tool betrachtet.

### Benutzerinterface

Für die Umsetzung wird Ruby [2], Ruby-Gnome [3], Ruby-Gems [4] und die Gnome Game Development Library [5] benutzt. Lehrer muss nicht ein paar Ruby-Platte und einen Abhängigkeitsinstallations:

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

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

- `gcc`
- `libgl1-mesa-dev`
- `libpng1.0-dev`
- `libxext-dev`
- `libxft-dev`
- `libxrender-dev`
- `ruby1.8-dev`

Die Paketbezeichnung startet von Debian/Ubuntu und kann/oder unter anderen Distributionen variieren. Unter Arch Linux gibt es beispielsweise keine extra dev-Pakete. Hier soll es Blick in die aktuelle Installationsanleitung [6] werfen.

Nach der Installation der Pakete installiert man die Gnome GUI Library mit `Ruby-Packer` über

```
gem install packer
```

Falls gefügt wird, werden `gnome-gui` installiert werden soll, wird man am besten `gem 8.7.14 (ruby)` aus. Falls eine Fehlermeldung der Art

```
ERROR: While executing gem ... (Gem::SourceNotFoundError)  
Could not find gem (~> 8) in any repository
```

kommt, sollte man die Installation einfach wiederholen, bis das Paket `gnome-gui` gefunden wird.

Nach der Kompilierung von `Gnome` kann man die Entwicklungsumgebung wieder installieren. Ruby & Co. müssen natürlich installiert bleiben.

### GDD-Struktur

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche für die Einstellungen des Spiems `robotz-gui-mlgpe` und zum anderen aus der Oberfläche für das Spiel selbst `robotz-gui`. Da die `robotz-gui` im `mlgpe` gestartet werden sollte, werden dessen Optionen hier nicht näher diskutiert.

```
bin/GDD-Struktur run mlgpe  
./././robotz-gui-mlgpe
```

auf. Im Standardfall kann man nun bereits auf „Spiel starten“ klicken und die Roboter-WI sollte den Roboter bewegen. Die Einstellungen im Spiel sind aber selbstverständlich. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.

Über „Neues Spiel starten“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei `last.txt` oder von der Startposition auf dem Spielbrett starten soll. Zusätzlich wird das Spielbrett und der Kartennamen nicht neu eingeleitet, wenn der Host nicht gesetzt ist. So kann man ein unterbrochenes Spiel fortsetzen.

Klickt man den Hosten bei „Spiel anhalten“ (auf der Roboter nicht über das Spielbrett), sondern das Spiel wirklich zum Ende beenden, der Roboter also das Ziel gefunden hat oder zerstört wurde. Erst dann sind die Abzüge beige/grün und man sieht den Roboter an der Endposition.

Mit „Spiel anhalten“ startet man das Spiel und kann den Roboter bei der Bewegung auszuwählen. Der Roboter hinterlässt ein transparentes Gitter Spur (Nicht, er verliert kein Öl), mit der man nachvollziehen kann, welche Felder er wie häufig betreten hat. Mit den Tasten **↑** und **↓** kann man die Bewegungsrichtung des Roboters verändern.

Die „grüne Spur“, d. h. die Bewegungen des Roboters werden während des Ablaufes in der Sequenzdatei abgespeichert, die im zugehörigen Feld angezeigt ist (im Standard ist die Datei `game.log.txt` eingetragen). Diese Datei hat beispielsweise folgenden Aufbau, wobei jede Zeile eine Bewegung oder Drehung angibt:

```
00  
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12
```

Die Datei kann später über „Spiel wiederholen“ geladen werden, sodass man die Bewegungen des Roboters erneut sieht. Über einen Klick auf den Eintrag „Pausieren“ kann man das Spiel kurz unterbrechen. „Anhalten“ startet das Spiel vor vorne bei einem zufälligen Kartennamen und ein neuer zufälliger `game.log.txt` und `last.txt` werden mit der GUI und `robotz-gui` werden in GUI gestartet. Hinweis: Die GUI hat eine Größe von 1024x768 Pixeln. Das heißt, um die sinnvoll nutzen zu können, sollte die Bildschirmauflösung größer als 1024x768 sein oder man muss im GUI-Fenster einstellen, dass das Programm im „Vollbildmodus“ starten soll. Dann werden die Grafiken auf eine kleinere Auflösung herunterskaliert.

### Die Spielregeln/Tricks

Robotz-GUI gibt es generell zwei Versionen zur Auswahl, die bei der Installation der Spielregeln installiert werden. Das Standardthema im Ordner `game` wurde von Arno Weiberg erstellt. Die Bilder unterliegen der GFDL [7].

Alternativ gibt es noch eine handgezeichnete Version im Ordner `game/pen`. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative Commons Lizenz. Namensnennung [Wagenführ](#) unter gleichem Bedingungen 3.0 Deutschland [8].

Dominik Harnel und  
Dominik Wagenführ

Seine Autoren sind Redakteure bei [heiseopen](#) und auch Programmierer aus Leidenschaft.

- [1] [heiseopen](#) Wettbewerb: [www.heiseopen.de/2008/04/01-programmierwettbewerb-gesamt](#)
- [2] [Ruby](#): [www.ruby-lang.org](#)
- [3] [Ruby-Gnome](#): [http://gnome.sourceforge.jp/de](#)
- [4] [RubyGems](#): [http://rubygems.org](#)
- [5] [Gnome Game Development Library](#): [http://libgame.org](#)
- [6] [Gnome Game Development Library](#): [http://gnome.sourceforge.jp/cgi-bin/install.html](#)
- [7] [GFDL](#): [http://www.gnu.org/licenses/licenses.html](#)
- [8] [CC-BY-SA](#): [http://creativecommons.org/licenses/by-sa/3.0/de/](#)

Abbildung: artikel8-layout2.pdf

# Mehrere Spalten

## Beispiel artikel8-layout3.tex

```
\usepackage{multicol}
\setlength{\columnsep}{0.5cm}
\begin{multicols*}{4}[\section{Visualisierungshilfe: ...}]
    TEXT
\end{multicols*}
```

- Paket `multicol`<sup>[12]</sup> für mehrspaltigen Satz
- Abstand zwischen zwei Spalten per `columnsep`
- Auch interessant: `columnrule` für Linie zwischen Spalten
- Angabe in `multicols`:
  - Anzahl Spalten (hier: 4)
  - optionales Argument: Befehle **vor** Start des mehrspaltigen Satzes
- `\begin{multicols*}` balanciert letzte Seite nicht aus

<sup>[12]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/multicol.html>



# Artikeldefinition

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Textgröße

Satzspiegel

Mehrere Spalten

Artikeldefinition

Kopf-/Fußleiste

Bilder

Parameter

## Beispiel artikel8-layout4a.tex

```
\usepackage{setspace}
\newcommand{\Autor}[1]
{%
    \begin{spacing}{1.2}%
        \begin{description}%
            \item[\textbf{Von:}] #1%
        \end{description}%
    \end{spacing}%
}
\newcommand{\Einleitung}[1]
{%
    \begin{spacing}{2}%
        \begin{flushleft}%
            {\huge #1}%
        \end{flushleft}%
    \end{spacing}%
    \vfill{}%
    \columnbreak{}%
}
```

- Paket `setspace`<sup>[13]</sup> für variablen Zeilenabstand
- Setzen per `\begin{spacing}{WERT}... \end{spacing}`
- Autor als `description`-Liste
- Einleitung als linksbündigen Flattertext (`flushleft`)
- `\vfill` füllt Spalte bis unten aus
- `\columnbreak` bricht Spalte um
- Anwendung:

```
\Autor{Dominik Honnef und\\Dominik Wagenführ}
\vskip-3em{ }%
\Einleitung{Da beim letzten Programmierwettbewerb
einige Nutzer nach einer grafischen Oberfläche (GUI)
fragten, können wir hier stolz eine präsentieren.}
```

<sup>[13]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/setspace.html>

## Visualisierungshilfe: Die GUI zum Programmierwettbewerb

**Von:** Dominik Honnef und  
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`

Ruby & Co. müssen natürlich installiert bleiben.

### GUI-START

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche für alle Einstellungen des Spiels `robots-gui-helper` und zum anderen aus der Oberfläche für das Spielbrett `robots-gui`. Da die `robots-gui` nie direkt gestartet werden sollte, werden dessen Optionen hier nicht näher erläutert.

Den GUI-Startler ruft man per

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „Spiel starten“ klicken und die Referenz-KI sollte den Roboter bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.

Über „Neues Spiel starten“ kann man einstellen, ob der Ro-

```
RU
RR
RR
MD
Z
```

Die Datei kann später über „Spiel wiederholen“ geladen werden, sodass man die Bewegungen des Roboters erneut sieht.

Über einen Klick auf den Eintrag „Pause“, kann man das Spiel kurz unterbrechen, „Neustart“ startet das Spiel von vorne (bei einem zufälligem Kartenstapel wird ein neuer zufälliger ausgewählt) und mit „Beenden“ verlässt man die GUI und landet wieder im GUI-Startler.

Hinweis: Die GUI hat eine Größe von 1024x768 Pixeln. Das heißt, um sie sinnvoll nutzen zu können, sollte die Bildschirmauflösung größer als 1024x768 sein oder man muss im GUI-Startler einstellen, dass das Programm im „Vollbildmodus“ starten soll. Dann werden

Abbildung: *artikel8-layout4a.pdf*

# Artikeldefinition

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Textgröße

Satzspiegel

Mehrere Spalten

Artikeldefinition

Kopf-/Fußleiste

Bilder

Parameter

## Beispiel artikel8-layout4b.tex

```
\newenvironment{Artikel}[4]
{%
    \pdfbookmark[2]{#2}{#1}
    \setcounter{linkcounter}{1}
    \begin{multicols}{4}[\section*{#2}]
        \begin{spacing}{1.05}
            \Autor{#3}
            \vskip-3em{ }%
            \Einleitung{#4}
        }
    }%
    \end{spacing}
\end{multicols}
}
```

- Artikel umfasst alle bisherigen Änderungen
- Lesezeichen für PDF-Betrachter mit pdfbookmark, hier der Ebene 2
- Titel per section\*, damit kein separates PDF-Lesezeichen gesetzt wird
- Anwendung:

```
\begin{Artikel}{2009_10_gui}{Visualisierungshilfe:  
Die GUI zum Programmierwettbewerb}{Dominik  
Honnef/\ \Dominik Wagenführ}{Da beim letzten  
Programmierwettbewerb einige Nutzer nach einer  
grafischen Oberfläche (GUI) fragten, können wir hier  
stolz eine präsentieren.}
```

## Visualisierungshilfe: Die GUI zum Programmierwettbewerb

Von: Dominik Honnet/  
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 Honnet 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`

Falls eine Fehlermeldung der Art

```
ERROR: While \
executing gem ... (\
Gem::\
GemNotFoundException)
Could not find \
gosu (> 0) in any\
repository
```

kommt, sollte man die Installation einfach wiederholen, bis das Paket gosu gefunden wird.

Nach der Kompilierung von Gosu kann man die Entwicklerpakete wieder deinstallieren. Ruby & Co. müssen natürlich installiert bleiben.

### GUI-START

Die grafische Oberfläche besteht aus zwei Teilen. Zum einen aus der Oberfläche für alle Einstellungen des Spiels `robots-gui-helper` und zum anderen aus der Oberfläche für das Spielbrett `robots-gui`. Da

die Anzeige freigeschaltet und man sieht den Roboter an der Endposition.

Mit „Spiel starten“ startet man das Spiel und kann dem Roboter bei der Bewegung zuschauen. Der Roboter hinterlässt ein transparente grüne Spur (Nein, er verliert kein Öl!), mit der man nachvollziehen kann, welche Felder er wie häufig betreten hat. Mit den Tasten  und  kann man die Bewegungsgeschwindigkeit des Roboters verändern.

Die „grüne Spur“, d. h. die Bewegungen des Roboters werden während des Ablaufes in der Sequenzdatei abgespeichert, die im zugehörigen Feld angegeben ist (als Standard ist die Datei `globalseq.txt` eingetragen). Diese Datei hat beispielsweise folgenden Aufbau, wobei jede Zeile einer Bewegung oder Drehung entspricht:

```
MU
```

Abbildung: *artikel8-layout4b.pdf*

## Beispiel artikel8-layout5a.tex

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhead{}
\fancyhead[LE]{Kategorie}
\fancyfoot{}
\fancyfoot[LE,RO]{\textbf{\textsc{\thepage}}}
\fancyfoot[RE,LO]{\textsl{SuperMAG XX/2009}}
\renewcommand*{\headrulewidth}{0.0pt}
\renewcommand*{\footrulewidth}{0.0pt}
```

- Paket `fancyhdr`<sup>[14]</sup> für eigene Kopf- und Fußleiste
- Anwahl über `\pagestyle{fancy}`
- `\fancyhead{}` und `\fancyfoot{}` ohne Argumente löscht Definitionen
- Argumente:
  - L/R/C – für Left/Right/Center (Links/Rechts/Mitte)
  - O/E – für Odd/Even (Ungerade/Gerade)
  - Auf geraden Seiten links oben (LE) steht die Kategorie
  - Auf geraden Seiten links unten und auf ungeraden rechts unten (LE,RO) steht die Seitenzahl
  - Auf geraden Seiten rechts unten und auf ungeraden links unten (RE,LO) steht die Ausgabe
- Linien bei Kopf- und Fußleiste werden über `headrulewidth` und `footrulewidth` ausgeschaltet

---

<sup>[14]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/fancyhdr.html>

um die Gosu-Datei 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.

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „Spiel starten“ klicken und die Referenz-KI sollte den Roboter bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.

Über „Neues Spiel starten“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei **bot.txt** oder von der Startposition auf dem Spielbrett starten soll. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn der Haken nicht gesetzt ist. So kann man ein unterbrochenes Spiel fortsetzen.

Entfernt man den Haken bei „Spiel animieren“, läuft der Roboter nicht über das Spielfeld, sondern das Spiel wird bis zum Ende berechnet, der Roboter also das Ziel gefunden hat oder zerstört wurde. Erst dann wird

```
RR
RR
MD
Z.
```

Die Datei kann später über „Spiel wiederholen“ geladen werden, sodass man die Bewegungen des Roboters erneut sieht.

Über einen Klick auf den Eintrag „Pause“, kann man das Spiel kurz unterbrechen, „Neustart“ startet das Spiel von vorne (bei einem zufälligem Kartenstapel wird ein neuer zufälliger ausgewählt) und mit „Beenden“ verlässt man die GUI und landet wieder im GUI-Starters.

Hinweis: Die GUI hat eine Größe von 1024x768 Pixeln. Das heißt, um sie sinnvoll nutzen zu können, sollte die Bildschirmauflösung größer als 1024x768 sein oder man muss im GUI-Starters einstellen, dass das Programm im „Vollbildmodus“ starten soll. Dann werden die Grafiken auf eine kleinere

## Beispiel artikel8-layout5b.tex

```
\newcommand{\SetFusszeile}[1]{%  
  \fancyfoot[RE,LO]{\textsl{SuperMAG #1}}  
  \hypersetup{pdftitle={SuperMag #1}}  
}
```

- Definition der Fusszeile
- Setzt auch PDF-Titel über `hypersetup`
- Anwendung (einmal am Anfang des Magazins):

```
\SetFusszeile{10/2009}
```

um die Gosu-Dev-Komponenten 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.

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „Spiel starten“ klicken und die Referenz-KI sollte den Roboter bewegen.

Die Einstellungen im Start sind alle selbsterklärend. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.

Über „Neues Spiel starten“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei **bot.txt** oder von der Startposition auf dem Spielbrett starten soll. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn der Haken nicht gesetzt ist. So kann man ein unterbrochenes Spiel fortsetzen.

Entfernt man den Haken bei „Spiel animieren“, läuft der Roboter nicht über das Spielfeld, sondern das Spiel wird bis zum Ende berechnet, der Roboter also das Ziel gefunden hat oder zerstört wurde. Erst dann wird

```
RR  
RR  
MD  
Z
```

Die Datei kann später über „Spiel wiederholen“ geladen werden, sodass man die Bewegungen des Roboters erneut sieht.

Über einen Klick auf den Eintrag „Pause“, kann man das Spiel kurz unterbrechen, „Neustart“ startet das Spiel von vorne (bei einem zufälligem Kartenstapel wird ein neuer zufälliger ausgewählt) und mit „Beenden“ verlässt man die GUI und landet wieder im GUI-Starter.

Hinweis: Die GUI hat eine Größe von 1024x768 Pixeln. Das heißt, um sie sinnvoll nutzen zu können, sollte die Bildschirmauflösung größer als 1024x768 sein oder man muss im GUI-Starter einstellen, dass das Programm im „Vollbildmodus“ starten soll. Dann werden die Grafiken auf eine kleinere

## Beispiel artikel8-layout5c.tex

```
\usepackage[absolute]{textpos}
\usepackage{rotating}
\fancyhead[LE]{\PrintHeader{Kategorie}}
\newcommand{\PrintHeader}[1]{%
  \begin{textblock}{1}(0,0)%
    \begin{sideways}%
      \setlength{\fboxsep}{3pt}
      \colbox{\maincolor}{\maintextcolor}{%
        {\Huge \textsl{\textbf{#1}}}}%
    }%
    \hspace*{1.5cm}%
  \end{sideways}%
\end{textblock}%
}
```

- Paket `textpos`<sup>[15]</sup> für absolute Positionierung von Inhalten
- Unterteilt eine Seite in ein Gitter mit Blöcken
- Option `absolute`: Blöcke werden absolut (meist) von oberer linker Ecke (0,0) angesprochen
- Setzt eine farbige Box mit der Kategorie in die obere linke Ecke (0,0) mit Blockbreite 1
- Über Paket `rotating`<sup>[16]</sup> und Befehl `sideways` Rotation der Box um 90 Grad

---

[15] <http://www.ctan.org/tex-archive/help/Catalogue/entries/textpos.html>

[16] <http://www.ctan.org/tex-archive/help/Catalogue/entries/textpos.html>

Auflösung herunterskaliert.

## DIE SPIELBRETT-THEMEN

In der GUI gibt es (derzeit) zwei Themen zur Auswahl, die die Gestaltung der Spielfelder beeinflussen.

Das Standardthema im Ordner **images/new** wurde von Arne Weinberg erstellt. Die Bilder unterliegen der GFDL [7].

Alternativ gibt es noch eine handgemalte Version im Ordner **images/pen**. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative-Commons-Lizenz „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“ [8].

Dominik Honnef und  
Dominik Wagenführ

Beide Autoren sind Redakteure bei freiesMagazin und auch Programmierer aus Leidenschaft.

## VERWEISE

- [1] freiesMagazin-Wettbewerb: [www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet](http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet)
- [2] Ruby: [www.ruby-lang.org/de](http://www.ruby-lang.org/de)
- [3] Ruby-Gnome2: [ruby-gnome2.sourceforge.jp/de](http://ruby-gnome2.sourceforge.jp/de)
- [4] RubyGems: [gems.rubyforge.org](http://gems.rubyforge.org)
- [5] Gosu Game Development Library: [www.libgosu.org](http://www.libgosu.org)
- [6] Ruby-Gnome2 Installation: [ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide](http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide)
- [7] GFDL: [www.gnu.org/licenses/fdl.html](http://www.gnu.org/licenses/fdl.html)
- [8] CC-BY-SA-3.0: [creativecommons.org/licenses/by-sa/3.0/de](http://creativecommons.org/licenses/by-sa/3.0/de)

Abbildung: *artikel8-layout5c.pdf*

## Beispiel artikel8-layout5d.tex

```
\newcommand*{\SetKategorie}[3]{%  
  \fancyhead{}%  
  \fancyhead[LE]{\PrintHeader{#1}}%  
  \pdfbookmark[1]{#1}{#1}%  
  \SetMainColor{#2}%  
  \SetMainTextColor{#3}%  
}
```

- Setzt die Kategorie in die Kopfleiste
- Setzt PDF-Lesezeichen der Ebene 1
- Setzt Farben
- Anwendung (einmal vor Beginn eines neuen Bereichs):

```
\SetKategorie{Programmierung}{mittelblau}{white}
```

Auflösung herunterskaliert.

## DIE SPIELBRETT-THEMEN

In der GUI gibt es (derzeit) zwei Themen zur Auswahl, die die Gestaltung der Spielfelder beeinflussen.

Das Standardthema im Ordner **images/new** wurde von Arne Weinberg erstellt. Die Bilder unterliegen der GFDL [7].

Alternativ gibt es noch eine handgemalte Version im Ordner **images/pen**. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der Creative-Commons-Lizenz „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“ [8].

Dominik Honnef und  
Dominik Wagenführ

Beide Autoren sind Redakteure bei freiesMagazin und auch Programmierer aus Leidenschaft.

## VERWEISE

- [1] freiesMagazin-Wettbewerb: [www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet](http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet)
- [2] Ruby: [www.ruby-lang.org/de](http://www.ruby-lang.org/de)
- [3] Ruby-Gnome2: [ruby-gnome2.sourceforge.jp/de](http://ruby-gnome2.sourceforge.jp/de)
- [4] RubyGems: [gems.rubyforge.org](http://gems.rubyforge.org)
- [5] Gosu Game Development Library: [www.libgosu.org](http://www.libgosu.org)
- [6] Ruby-Gnome2 Installation: [ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide](http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide)
- [7] GFDL: [www.gnu.org/licenses/fdl.html](http://www.gnu.org/licenses/fdl.html)
- [8] CC-BY-SA-3.0: [creativecommons.org/licenses/by-sa/3.0/de](http://creativecommons.org/licenses/by-sa/3.0/de)

Abbildung: artikel8-layout5d.pdf

## Beispiel artikel9-bilder1.tex

```
\newcommand{\fcolbox}[4]{%
  \fcolorbox{#1}{#2}{\textcolor{#3}{#4}}}%
}
\newcommand{\Bildunterschrift}[3]
{%
  \setlength{\fboxsep}{5pt}%
  \setlength{\fboxrule}{0.8pt}%
  \noindent{%
    \fcolbox{black}{dunkelgrau}{white}{%
      \parbox{#1-2\fboxsep-2\fboxrule}{%
        \begin{flushleft}%
          \vskip-5pt{%
            \textbf{\MakeUppercase{#2}} | #3%
          }
        \end{flushleft}%
        \vskip-5pt{%
          }%
        }%
      }%
    }%
  }%
}
```

- Für bessere Kontrolle umschalten auf `multicols*` in Artikeldefinition
- Definition `\fcolbox` für farbige Box mit farbigem Rahmen über `\fcolorbox`
- Erster Teil der Bildunterschrift in Großbuchstaben (`\MakeUppercase`)
- `\Bildunterschrift` als dunkelgraue Box mit weißer Schrift
- Text als linkbündiger Flattertext (`flushleft`), kein Blocksatz
- Aufgrund des Flattertextes zu hohe Box, daher Kompensation durch `\vskip`

## Beispiel artikel9-bilder1.tex

```
\usepackage{graphicx}
\newcommand{\BildInternal}[4][\linewidth]
{%
    \includegraphics[width=#1]{Bilder/#2}\[-2px]
    \Bildunterschrift{#1}{#3}{#4}\
}
\newcommand{\Bild}[4][\linewidth]
{%
    \noindent{ }%
    \parbox{#1}{\BildInternal[#1]{#2}{#3}{#4}}%
}
```

- Paket `\graphicx`<sup>[17]</sup> zur Einbindung von Grafiken
- `\Bildinternal` wird später benötigt
- Darstellung des Bildes mit `\includegraphics`
- Kompensation von `-2px`, damit Bildunterschrift direkt am Bildrand
- Eigentliches `\Bild` ohne Einrückung
- Achtung: Nicht mit prozentualem `\linewidth` bei Bildbreite arbeiten!
  - Führt ggf. zu Potenzierung der Skalierung, d. h. aus halber Zeilenbreite (`0.5\linewidth`) wird ein viertel Zeilenbreite

---

<sup>[17]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/graphicx.html>

- Alle Bilder im Text setzen mit

```
Viel Text ... \\
```

```
\Bild{gui.png}{Auf Irrwegen}{Dumme KIs irren etwas  
länger herum.}\\
```

- Vor und nach jedem Bild Abstand per \\



## Beispiel artikel9-bilder2.tex

```
\newlength{\VSkipHeight}
\newcommand{\BildAbs}[3]
{%
  \begin{textblock}{1}(0,0)%
    \Bild[\paperwidth]{#1}{#2}{#3}%
  \end{textblock}%
  \settoheight{\VSkipHeight}{%
    \BildInternal[\paperwidth]{#1}{#2}{#3}%
  }%
  \addtolength{\VSkipHeight}{-3em}%
  \vspace*{\VSkipHeight}%
}
```

- Bild absolut positioniert per `textblock` über ganze Seitenbreite (`\paperwidth`)
- Problem: Text durchläuft Bild
- Lösung: Abstand in Höhe des Bildes
  - Neue Höhe `\VskipHeight`, die über `\settoheight` Höhe des Bildes annimmt
  - 3em Abzug zur Kompensation
  - Abstand per `\vspace*` anwenden (ohne Stern wirkt es nicht am Seitenanfang)
- Anwendung auf erstes Bild:

```
\BildAbs{gui2.png}{Der Weg ist das Ziel}{Ein schlauer  
Roboter sucht den direkten Weg zum Ziel.}
```



## Beispiel artikel9-bilder3.tex

```
\newcommand{\sbreak}[1] [-1em]{\linebreak[4] \\ [#1]}  
\newcommand{\cbreakfill}{\vfill\columnbreak}  
\newcommand{\csbreakfill}{\sbreak\cbreakfill\noindent}
```

- Neue Umbrüche:
  - `\sbreak` – für „sanften“ Zeilenumbruch, streckt Zeile bis zum Zeilenende mit `\linebreak`
  - `\cbreakfill` – beendet Spalte, füllt diese vorher bis zum Ende mit Leerraum
  - `\csbreakfill` – Kombination aus beiden, falls Spalte innerhalb eines Satzes umgebrochen werden muss

- Anwendung auf zweites Bild:

```
\Bild[2\columnwidth+\columnsep]{helper.png}{Hilfe ist  
unterwegs}{Der GUI-Helfer startet die Spiele-GUI.}\\  
...  
Bewegung zuschauen. Der Roboter  
\csbreakfill{}%  
hinterlässt ein transparente grüne Spur
```



## Beispiel artikel9-bilder4.tex

```
\usepackage{wrapfig}
\newcommand{\Einschub}[3]
{%
  \begin{wrapfigure}[#1]{#2}{#3}
  \vfill
  \end{wrapfigure}
}
```

- Paket `wrapfig`<sup>[18]</sup> für textumflossene Bereiche
- `Einschub`, der Freiraum mit `wrapfigure` lässt, Argumente:
  - 1 Höhe des Einschubs als Anzahl Zeilen
  - 2 Ausrichtung: `l` (links) oder `r` (rechts)
  - 3 Breite des Einschubs
- `\vfill` als leere „Füllung“

<sup>[18]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/wrapfig.html>

- Freilassung von Bereichen geht **nicht** automatisch!
- Anwendung auf drittes Bild:

```
\Bild[1.6\columnwidth]{gui.png}{Auf Irrwegen}{Dumme  
KIs irren etwas länger herum.}\ [0.1em]  
...  
sieht den Roboter an der Endposition.  
\cbreakfill{}%  
  
\Einschub{16}{1}{1.8cm}%  
Mit \Menu{Spiel starten} startet man das ...
```



## Beispiel artikel9-bilder5.tex

```
... Kartenstapel wird ein neuer\sbreak[-1.5em]

\begin{Quellen}[2\columnwidth+\columnsep]
...
\end{Quellen}

\noindent{}%
zufälliger ausgewählt) und mit \Menu{Beenden} ...
```

- Quellen über zwei Spalten ...
- ... und an andere Stelle verschoben
- Für Konvertierung in andere Formate (z. B. HTML) wegen Absätzen/Leerzeilen kein gutes Vorgehen!



# Optionale Parameter

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Bilder

Parameter

## Beispiel artikelA-xkeyval.tex

```
\usepackage{xkeyval}
\newlength{\BildImageWidth}
\newboolean{BildAbsolutePos}
\makeatletter
\define@key{Bild}{width}{\setlength{\BildImageWidth}{#1}}
\define@boolkey{Bild}{absolut}[true]{%
  \setboolean{BildAbsolutePos}{#1}%
}
\makeatother
```

# Optionale Parameter

- Neue Länge `\BildImageWidth` für Breite des Bildes
- Boolescher Wert `BildAbsolutePos`, ob Bild absolut und über ganze Seitenbreite dargestellt werden soll
- Paket `xkeyval`<sup>[19]</sup> zur Auswertung von optionalen Parametern
- Neuen Standardschlüssel definieren mit `\define@key`:
  - Gruppe (hier Bild)
  - Schlüsselname (hier `width`)
  - Befehl, Argument per #1 ansprechbar
- Neuen Boolescher Schlüssel definieren mit `\define@boolkey`:
  - kennt nur `true` und `false`
  - Gruppe (wieder Bild)
  - Schlüsselname (hier `absolut`)
  - Standardwert, wenn nur Parameter ohne Zuweisung angegeben

---

<sup>[19]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/xkeyval.html>

# Optionale Parameter

## Beispiel artikelA-xkeyval1.tex (Fortsetzung)

```
\usepackage{xifthen}
\newcommand*{\CalcBildValues}[1]
{%
  \presetkeys{Bild}{width=\linewidth,absolut=false}{}%
  \setkeys{Bild}{#1}%
  \ifthenelse{\boolean{BildAbsolutePos}}%
    {\setlength{\BildImageWidth}{\paperwidth}}{}%
}
```

# Optionale Parameter

- Auswertung der Parameter in Kommando `\CalcBildValues`
- `\setkeys` wertet Schlüssel einer Gruppe aus
- `\presetkeys` belegt Parameter vor:
  - Gruppe
  - Vorbelegungen vor `\setkeys`
  - Vorbelegungen nach `\setkeys`
  - Kommt Schlüssel in `\setkeys`, keine Vorbelegung!
- Paket `xifthen`<sup>[20]</sup> für erweiterte if-Abfragen
- Wenn absolute Positionierung, dann Bildgröße fest auf Seitenbreite

---

<sup>[20]</sup><http://www.ctan.org/tex-archive/help/Catalogue/entries/xifthen.html>

## Beispiel artikelA-xkeyval1.tex (Fortsetzung)

```
\newcommand{\BildInternal}[3] [\linewidth]
{%
  \includegraphics [width=\BildImageWidth]%
    {Bilder/#1}\ [-2px]
  \Bildunterschrift{\BildImageWidth}{#2}{#3}\
}
```

- `\BildInternal` ohne optionalen Parameter, da Bildbreite `\BildImageWidth` global verfügbar

# Optionale Parameter

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Bilder

Parameter

## Beispiel artikelA-xkeyval1.tex

```
\newcommand{\Bild}[4] []
{%
  \CalcBildValues{#1}%
  \ifthenelse{\boolean{BildAbsolutePos}}
  {%
    \begin{textblock}{1}(0,0)%
  }{}%
  \noindent{}%
  \parbox{\paperwidth}{\BildInternal{#2}{#3}{#4}}%
  \ifthenelse{\boolean{BildAbsolutePos}}
  {%
    \end{textblock}%
    \settoheight{\VSkipHeight}{\BildInternal{#2}{#3}{#4}}%
    \addtolength{\VSkipHeight}{-3em}%
    \vspace*{\VSkipHeight}%
  }{}%
}
```

- Auswertung der Schlüssel in `CalcBildValues`
- `\BildAbs` fällt weg, daher Zusammenfügen des Codes in `\Bild`
- Wenn `BildAbsolutePos` gesetzt ist, dann absolute Positionierung und Verschieben des Textes um Bildhöhe wie zuvor



# Optionale Parameter

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Bilder

Parameter

## Beispiel artikelA-xkeyval2.tex

```
\newlength{\BildBoxWidth}  
\makeatletter  
\define@key{Bild}{boxwidth}{\setlength{\BildBoxWidth}{#1}}  
\makeatother
```

- Neue Länge `\BildBoxWidth` für Breite der Bildunterschrift
- Neuer Schlüssel `boxwidth` für Zuweisung der Breite

# Optionale Parameter

## Beispiel artikelA-xkeyval2.tex (Fortsetzung)

```
\newcommand*{\CalcBildValues}[1]
{%
  \presetkeys{Bild}{width=\linewidth,absolut=false,%
    boxwidth=Opt}{}%
  \setkeys{Bild}{#1}%
  \ifthenelse{\boolean{BildAbsolutePos}}
  {%
    \setlength{\BildImageWidth}{\paperwidth}%
  }{}%
  \ifdim\BildBoxWidth=Opt{}%
    \setlength{\BildBoxWidth}{\BildImageWidth}%
  \fi{}%
}
```

# Optionale Parameter

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Bilder

Parameter

- Vorbelegung von `boxwidth` mit Wert `0pt`
- Wenn Wert am Ende immer noch `0pt`, dann Box so breit wie das Bild machen
- Wieso so umständlich?
- Wenn `boxwidth=\linewidth` als Vorbelegung hat, dann wäre die Box immer `\linewidth` groß, auch wenn Bildbreite `width=2\linewidth` gesetzt ist.

## Beispiel artikelA-xkeyval2.tex (Fortsetzung)

```
\newcommand\BildInternal[3]
{%
  \includegraphics[width=\BildImageWidth]%
    {Bilder/#1}\[-2px]
  \Bildunterschrift{\BildBoxWidth}{#2}{#3}\
}
```

- `\BildInternal` erhält `\BildBoxWidth` als Breite der Bildunterschrift



# Optionale Parameter

L<sup>A</sup>T<sub>E</sub>X-Workshop

Dominik  
Wagenführ

Einleitung

Textformat

Layout

Bilder

Parameter

## Beispiel artikelA-xkeyval3.tex

```
\newcommand*\BildBoxAlign{}
\makeatletter
\define@choicekey{Bild}{boxalign}[\val\ba]{left,right}
{%
  \ifcase\ba\relax
    \renewcommand*\BildBoxAlign{}
  \or
    \renewcommand*\BildBoxAlign{\hspace*{\fill}}
  \fi
}
\makeatother
```

- Neues Kommando `\BildBoxAlign` für Ausrichtung der Bildunterschrift
- `\define@choicekey` als Schlüssel mit festen Zuweisungsmöglichkeiten:
  - Gruppe (immer noch Bild)
  - Schlüsselname (hier `boxalign`)
  - Zuweisung des Wertes in `\ba`
  - Liste der möglichen Zuweisungen
  - Unterscheidung mit `case`-Anweisung, was zu tun ist:
    - `left` – gar nichts, daher ist `\BildBoxAlign` leer
    - `right` – Zeile auffüllen bis Ende
- Reihenfolge in `case`-Anweisung so wie in Liste definiert

## Beispiel artikelA-xkeyval3.tex (Fortsetzung)

```
\presetkeys{Bild}{width=\linewidth,absolut=false,%  
                boxwidth=0pt,boxalign=left}{}%
```

- Neue Vorbelegung für boxalign in `\CalcBildValues`

## Beispiel artikelA-xkeyval3.tex (Fortsetzung)

```
\newcommand{\BildInternal}[3]
{%
  \includegraphics[width=\BildImageWidth]%
    {Bilder/#1}\[-2px]
  \BildBoxAlign\Bildunterschrift{\BildBoxWidth}{#2}{#3}\
}
```

- BildInternal wird um \BildBoxAlign erweitert:
  - Wenn boxalign=left, passiert nichts
  - Wenn boxalign=right, wird Bildunterschrift an rechten Rand geschoben



Vielen Dank für die Aufmerksamkeit  
und viel Spaß beim Nachmachen!

Folien und Beispiele unterliegen der Creative Commons  
„Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland“.



Copyright 2009 Dominik Wagenführ