Introduction to presentations with `beamer`

Thomas Thurnherr

Abstract

The document class `beamer` provides flexible commands to prepare presentations with an appealing look. Here I introduce the basics of the `beamer` class intended for new users with a basic knowledge of LaTeX. I cover a range of topics from how to create a first slide to dynamic content and citations.

1 Introduction

The LaTeX document class `beamer` [1] was written to help with the creation of presentations held using a projector. In German, the English word *Beamer* describes a projector, which is likely the reason for Till Tantau, the package author, to choose this particular name. Many macros available in the standard LaTeX document classes are used in `beamer`, although sometimes the result might look different. The `beamer` package comes with extensive documentation, which is included in most TeX distributions (such as TeX Live) and available online on CTAN. As with other document classes, the output document is likely in portable document format (PDF). This imposes certain limitations on animations well-known from commercial software. However, it has always been a strength of (La)TeX to let the author focus on the content, and `beamer` extends this concept to slide presentations.

2 The very basics

To create a presentation, we set the document class to `beamer`:

```
\documentclass{beamer}
```

The main difference between standard LaTeX document classes and `beamer` is that content does not continuously "flow" across multiple pages, but is limited to a single slide. The environment name `frame` is used for slides, usually to produce a single slide (sometimes several, but we will get to that later). A `frame` contains a title and a body. Furthermore, at the bottom of every `frame`, `beamer` automatically adds a navigation menu.

```
\begin{frame}
  \frametitle{Slide title}
  %Slide body
\end{frame}
```

3 In the preamble

As with the standard LaTeX document classes, the preamble serves to load packages, define the content of the title slide, and alter the appearance of the presentation.

3.1 Presentation title

`Beamer` reuses the standard LaTeX macros to create the title page: `\title`, `\author`, and `\date`.

```
\title{Beamer presentation title}
\author{Presenter's name}
\date{\today}
```

We use these further below to create a title page frame.

3.2 Presentation appearance

In `beamer`, "themes" change the appearance of a presentation. Themes define the style and the color of a presentation. By default, `beamer` loads the rather bland `default` theme. To change the theme to something more appealing, we can use the following command in the preamble with a theme name we like:

```
\usetheme{default} % default theme
```

There are a great number of themes distributed with LaTeX. They are often named after cities. Try for example: `Berkeley`, `Madrid`, or `Singapore`, to name a few (figure 1). Also, look for `beamer` theme galleries online.

4 Presentation slides

4.1 Creating a basic frame

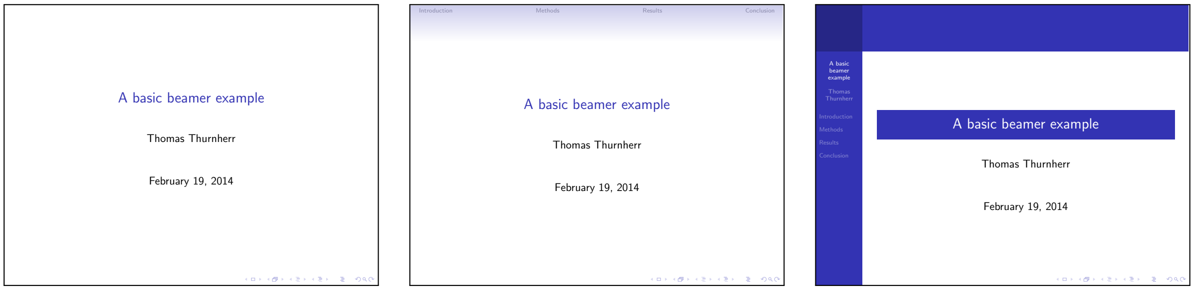
A `frame` may contain a number of different things, including simple text, formulas, figures, tables, etc. Most often, however, a `frame` contains numbered or bulleted lists. To create lists, we use the standard list environments: `enumerate` and `itemize`. An example of a bulleted list is shown below:

```
\begin{frame}
  \frametitle{List types in \LaTeX}
  \begin{itemize}
    \item Bulleted list: itemize
    \item Numbered list: enumerate
    \item Labeled list: description
  \end{itemize}
\end{frame}
```

Although the result looks different, lists work in the same way as in other document classes; they can be nested and customized to your needs.

4.2 Title page frame

We have already seen how to define a title in the preamble. Now we want to use this title to create a title page frame.



A basic beamer example

Thomas Thurnherr

February 19, 2014

Figure 1: Title page frames for beamer themes: `default`, `Singapore`, and `Berkeley`.

```
\begin{frame}
  \titlepage
  % alternatively \maketitle can be used
\end{frame}
```

4.3 Table of contents

To add structure to a presentation and create an outline, we can use `\section` and `\subsection`, together with `\tableofcontents`. These commands are used outside of frames. To create an outline at the beginning of a presentation, we use:

```
\section{Presentation Outline}
\begin{frame}
  \frametitle{Outline}
  \tableofcontents
\end{frame}
```

For long presentations, it may make sense to show the outline again at the beginning of a new `\section`. We use the option `currentsection` to `\tableofcontents` to highlight the current section.

```
\section{New Section Title}
\begin{frame}
    \frametitle{Outline}
    \tableofcontents[currentsection]
  \end{frame}
```

4.4 Adding figures

Frames are single entities and therefore figures and tables do not need to be wrapped in their respective floating environments. Also, we probably do not wish to add a caption. Therefore, to show a figure, `\includegraphics` with appropriate alignment and scaling (see the `graphicx` package [3]) is sufficient:

```
\begin{frame}
  \frametitle{Adding a figure to a frame}
  \centering
  \includegraphics[width=0.8\linewidth]
    {some-figure-file}
\end{frame}
```

Similarly with tables, we omit the `table` environment and directly use `tabular`.

4.5 Defining multiple columns

By default, content is stacked vertically. Therefore, if you have a list, then a figure, then a text paragraph, first the list is produced, with the figure below and lastly the text. Often, it's desirable to position content next to each other. There are two (identical) methods to split a slide horizontally into two or more columns: the `minipage` environment and the `beamer`-specific `columns` environment.

```
\begin{frame}
  \frametitle{Two column example: minipage}
  \begin{minipage}{0.48\linewidth}
    % content column 1
  \end{minipage}
  \quad % adds some whitespace
  \begin{minipage}{0.48\linewidth}
    % content column 2
  \end{minipage}
\end{frame}

\begin{frame}
  \frametitle{Two column example: columns}
  \begin{columns}
    \begin{column}{0.48\linewidth}
      % content column 1
    \end{column}
    \quad
    \begin{column}{0.48\linewidth}
      % content column 2
    \end{column}
  \end{columns}
\end{frame}
```

5 Simple animations

It may be an exaggeration to use the word "animations". What I will show is merely how to add, remove and replace parts of the content, primarily text. However, I believe this is good enough to keep the audience interested, everything else is just a distraction.

5.1 Add items dynamically

The `beamer` command `\pause` adds content gradually, by pausing and waiting for the presenter to press a button. For example, we can use `\pause` in a list to reveal one item after another. You might wonder how this is possibly translated into a PDF. There is really no magic to it; LaTeX just produces three slides with the same page number, adding an extra item one each subsequent slide. Try for yourself:

```
\begin{frame}
  \frametitle{Usage of pause}
  \begin{itemize}
    \item First item, shown with the slide
      \pause
    \item Next item, revealed after pressing
      a button
      \pause
    \item Last item, revealed after pressing
      a button again
  \end{itemize}
\end{frame}
```

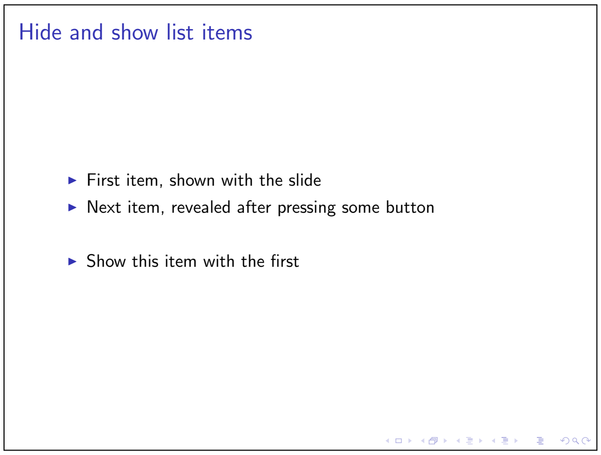
5.2 Hide and show content

“Overlays” is a slightly more sophisticated concept. Overlays use pointed brackets to hide, reveal and overwrite content. For example, the specification `\item<1->` means: “from slide 1 on” (see figure 2).

```
\begin{frame}
  \frametitle{Hide and show list items}
  \begin{itemize}
    \item<1-> First item, shown with the slide
    \item<2-> Next item, revealed after
      pressing some button
    \item<3-> Last item, revealed again after
      pressing some button
    \item<1-> Show this item with the first
  \end{itemize}
\end{frame}
```

We can also combine ranges of numbers. Assuming more than 7 overlays, to show an item on all but slides 3 and 6, we use: `\item<-2,4-5,7->`. Items always occupy their space, even if they are not shown. Joseph Wright’s article elsewhere in this issue provides more examples [5].

This syntax works with other content too, as implemented in the commands `\uncover` and `\only`. The difference between them is that `\uncover` occupies space when hidden, whereas `\only` does not, and can therefore be used to overwrite previous content.



Hide and show list items

- First item, shown with the slide
- Next item, revealed after pressing some button

- Show this item with the first

Figure 2: Hide and show list items.

```
\begin{frame}
  \frametitle{Hide and show content}
  \uncover<1> { % adds content }
  \uncover<2> { % add additional content }
\end{frame}

\begin{frame}
  \frametitle{Hide and overwrite content}
  \only<1> { % adds content }
  \only<2> { % replaces previous content }
\end{frame}
```

Similar to the `itemize` example above, much more sophisticated overlays can be created using `\uncover` and `\only`.

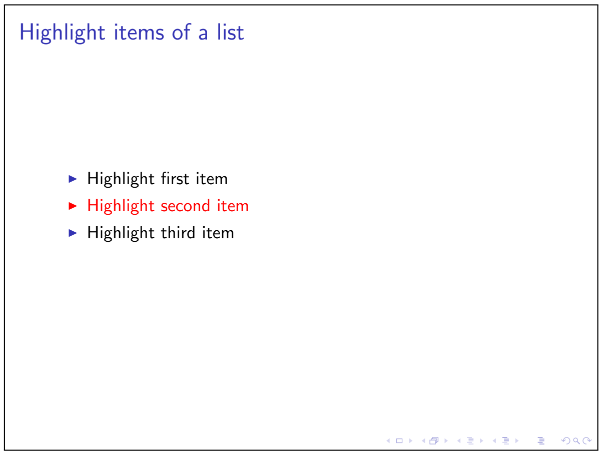
5.3 Highlighting items

Besides hiding and revealing, we can also highlight text upon a button press. In `beamer`, this is called an `alert` (see figure 3):

```
\begin{frame}
  \frametitle{Highlight items of a list}
  \begin{itemize}
    \item<alert@1> Highlight first item
    \item<alert@2> Highlight second item
    \item<alert@3> Highlight third item
    \item<4- | alert@4> Combine reveal and
      highlight
  \end{itemize}
\end{frame}
```

6 Citations and bibliography

You can generate a bibliography the same way as with a standard LaTeX document class. Personally, I prefer to show the bibliography entries on the same slide where they are cited. I use the `biblatex` package [2]



Highlight items of a list

- Highlight first item
- Highlight second item
- Highlight third item

Figure 3: Highlight list items.

for this, which provides a variety of methods. The commands `\footcite` and`\footfullcite` produce references according to the style (e.g. `authoryear`), and full references respectively. I use an external BibTeX file to store references. Here is an example.

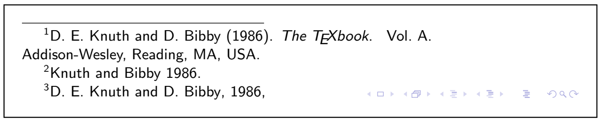
```
% Preamble
\usepackage[backend=biber, maxnames=2,
  firstinits=true,style=authoryear]{biblatex}
\bibliography{path/to/references.bib}

% Citations
\begin{frame}
  \frametitle{Citing other people's work}
  \begin{itemize}
    \item Full citation
      \footfullcite{knuth86}
    \item Author-year citation
      \footcite{knuth86}
  \end{itemize}
```

If your preferred style is not available, you can define your own citation style using the `biblatex` command `\DeclareCiteCommand`. The invocation below prints references as footnotes, showing the author, year, and journal title.

```
% Preamble
\usepackage[backend=biber, maxnames=2,
  firstinits=true]{biblatex}
\bibliography{path/to/references.bib}
\DeclareNameAlias{sortname}{first-last}
\DeclareCiteCommand{\footcustomcite}{}{%
    \footnote{\printnames[author]{author},
      \printfield{year},
      \printfield{journaltitle}
      \printfield{booktitle}}}{;}{}

% Citations
\footcustomcite{knuth86}
```



[1]D. E. Knuth and D. Bibby (1986). *The TEXbook*. Vol. A. Addison-Wesley, Reading, MA, USA.
[2]Knuth and Bibby 1986.
[3]D. E. Knuth and D. Bibby, 1986,

Figure 4: Full, `authoryear`-style, and custom citation of *The TEXbook*.

Figure 4 shows a citation (for *The TEXbook*) in the full, `authoryear`, and above custom styles.

7 Creating handouts

If you teach a class or give a talk it might be appropriate to provide handouts. The `beamer` document class option `handout` reduces overlays to a single slide and removes the navigation bar at the bottom. In addition, we can combine multiple slides to a single physical page, using the `pgfpages` package [4]:

```
\documentclass[handout]{beamer}
\usepackage{pgfpages}
\pgfpagesuselayout{4 on 1}[a4paper,
  border shrink=5mm, landscape]
```

References

[1] beamer — A LaTeX class for producing presentations and slides. `http://www.ctan.org/pkg/beamer`. Accessed: 2014-02-19.

[2] biblatex — Bibliographies in LaTeX using BibTeX for sorting only. `http://www.ctan.org/pkg/biblatex`. Accessed: 2014-02-19.

[3] graphicx — Enhanced support for graphics. `http://www.ctan.org/pkg/graphicx`. Accessed: 2014-02-19.

[4] pgf — Create PostScript and PDF graphics in TeX. `http://www.ctan.org/pkg/pgf`. Accessed: 2014-02-19.

[5] Joseph Wright. The `beamer` class: Controlling overlays. *TUGboat*, 35(1):31–33, 2014.

◇ Thomas Thurnherr
texblog (at) gmail dot com
http://texblog.org