BEAMER

An Introduction

Ryan Siskind

NCSU

October 3, 2007

Outline

1. Introduction
   - Overview of the Beamer Class
   - Basic Setup
2. Creating a Slideshow
   - Themes and Colors
   - Titles
   - Sections
3. Overlays
4. Graphics
   - Figures
   - Movies

History of Beamer

"I originally created BEAMER *mainly in my spare time as a small private collection of macros to make using the* `seminar` *class easier. The first full version was for my PhD defense presentation in February 2003. A month later, I put the package on CTAN at the request of some colleagues. After that, things somehow got out of hand."*
-Till Tantau, 2004

Features of the Beamer Class

1. Can use standard `pdflatex` and `latex+dvips`.

Features of the Beamer Class

1. Can use standard `pdflatex` and `latex+dvips`.
2. Normal LaTeX commands;
   - uses section/subsection/etc. for structuring;
   - commands such as `\tableofcontents` and `align` have the same meaning as in `article` class

Features of the Beamer Class

1. Can use standard `pdflatex` and `latex+dvips`.
2. Normal LaTeX commands;
   - uses section/subsection/etc. for structuring;
   - commands such as `\tableofcontents` and `align` have the same meaning as in `article` class
3. Easy overlays (which is what is going on right now)

Features of the Beamer Class

1. Can use standard `pdflatex` and `latex+dvips`.
2. Normal LaTeX commands;
   - uses section/subsection/etc. for structuring;
   - commands such as `\tableofcontents` and `align` have the same meaning as in `article` class
3. Easy overlays (which is what is going on right now)
4. No external programs needed other than what you already use for LaTeX

Features of the Beamer Class

1. Can use standard `pdflatex` and `latex+dvips`.
2. Normal LaTeX commands;
   - uses section/subsection/etc. for structuring;
   - commands such as `\tableofcontents` and `align` have the same meaning as in `article` class
3. Easy overlays (which is what is going on right now)
4. No external programs needed other than what you already use for LaTeX
5. Font manipulation, movie files, fun stuff, etc.

CTAN

BEAMER is hosted at

http://latex-beamer.sourceforge.net/

CTAN

BEAMER is hosted at

http://latex-beamer.sourceforge.net/

1. Files (BEAMER, pgf, xcolor);
2. Instructions for installation;
3. BEAMER examples.

Installation

1. Go to http://latex-beamer.sourceforge.net/ or http://sourceforge.net/projects/latex-beamer/
2. Choose to download the “LaTeX-Beamer Class”
3. Download the latest version of `latex-beamer`, `pgf`, and `xcolor`

Important Note

The Beamer user guide is found in the `latex-beamer` file and is ÜBER helpful.

More Installation

For a permanent installation:

1. Find your local `texmf` tree (usually found in `/usr/local/share/texmf/`, `c:\localtexmf\`, or `c:\Program Files\TeXLive\texmf-local\`)

More Installation

For a permanent installation:

1. Find your local `texmf` tree (usually found in `/usr/local/share/texmf/`, `c:\localtexmf\`, or `c:\Program Files\TeXLive\texmf-local\`)
2. In the `texmf` directory, create the sub-sub-sub-directories;
   - `texmf/tex/latex/beamer`,
   - `texmf/tex/latex/pgf`, and
   - `texmf/tex/latex/xcolor`

More Installation

For a permanent installation:

1. Find your local `texmf` tree (usually found in `/usr/local/share/texmf/`, `c:\localtexmf\`, or `c:\Program Files\TeXLive\texmf-local\`)
2. In the `texmf` directory, create the sub-sub-sub-directories;
   - `texmf/tex/latex/beamer`,
   - `texmf/tex/latex/pgf`, and
   - `texmf/tex/latex/xcolor`
3. Place all UNZIPPED files from the packages you already downloaded into these new directories.

More Installation

For a permanent installation:

1. Find your local `texmf` tree (usually found in `/usr/local/share/texmf/`, `c:\localtexmf\`, or `c:\Program Files\TeXLive\texmf-local\`)
2. In the `texmf` directory, create the sub-sub-sub-directories;
   - `texmf/tex/latex/beamer`,
   - `texmf/tex/latex/pgf`, and
   - `texmf/tex/latex/xcolor`
3. Place all UNZIPPED files from the packages you already downloaded into these new directories.
4. Rebuild the TEXfile database by running the command `texhash`, `mktexlsr`, or via menu options (if available)

My First Slide

```
\documentclass{beamer}

\begin{document}

  \begin{frame}
    Hello World!
  \end{frame}

\end{document}
```

Presenting in Style

Themes dictate colors, information bars, and layout of presentation. This presentation uses the theme `\usetheme{CambridgeUS}`

- Themes, p135-148;
- Templates, p149-158;
- Colors, p162-175.

Frame Titles

...and Subtitles

2 ways to create titles and subtitles for a frame:

1. `\begin{frame}{`*Frame Title*`}{`*Frame Subtitle*`}`
2. `\frametitle{`*Frame Title*`}\framesubtitle{`*Frame Subtitle*`}`

Sectioning

Notice the sections and subsections at the top of each slide.

- `\section[`*Short Section Name*`]{`*Long Section Name*`}`
- `\subsection[`*Short Subsection Name*`]{`*Long Subsection Name*`}`

Sectioning

Notice the sections and subsections at the top of each slide.

- `\section[`*Short Section Name*`]{`*Long Section Name*`}`
- `\subsection[`*Short Subsection Name*`]{`*Long Subsection Name*`}`

“Short names” go into slide headers;
“Long names” go into outlines.

Sectioning

Notice the sections and subsections at the top of each slide.

- `\section[`*Short Section Name*`]{`*Long Section Name*`}`
- `\subsection[`*Short Subsection Name*`]{`*Long Subsection Name*`}`

"Short names" go into slide headers;
"Long names" go into outlines.

All sections and subsections automatically added to slideshow outline!

Looooooong Slides

BEAMER does not automatically put what doesn't fit from one slide onto another slide.

- You must keep track of slide lengths yourself; or
- you can use the frame option
  `\begin{frame}[allowframebreaks]`

This automatically breaks up the long slide and puts the extra content onto new slides.

Looooooong Slides

BEAMER does not automatically put what doesn't fit from one slide onto another slide.

- You must keep track of slide lengths yourself; or
- you can use the frame option

  ```
  \begin{frame}[allowframebreaks]
  ```

This automatically breaks up the long slide and puts the extra content onto new slides.

+ You don't have to worry about the length of your slides.

Loooooong Slides

BEAMER does not automatically put what doesn't fit from one slide onto another slide.

- You must keep track of slide lengths yourself; or
- you can use the frame option
  `\begin{frame}[allowframebreaks]`

This automatically breaks up the long slide and puts the extra content onto new slides.
+ You don't have to worry about the length of your slides.
+ Slide title is continued on each subsequent slide from the original frame.

Loooooong Slides

BEAMER does not automatically put what doesn't fit from one slide onto another slide.

- You must keep track of slide lengths yourself; or
- you can use the frame option `\begin{frame}[allowframebreaks]`

This automatically breaks up the long slide and puts the extra content onto new slides.
+ You don't have to worry about the length of your slides.
+ Slide title is continued on each subsequent slide from the original frame.
- Most overlay options are not usable.

Why and How Overlays Are Used

- Much like the transitions in PowerPoint
- Allows different information to be shown at different times on same slide
- User defines when information is shown using <*Transparency numbers*>

Why and How Overlays Are Used

- Much like the transitions in PowerPoint
- Allows different information to be shown at different times on same slide
- User defines when information is shown using <*Transparency numbers*>

If you want information to show up immediately: `<1->`

Why and How Overlays Are Used

- Much like the transitions in PowerPoint
- Allows different information to be shown at different times on same slide
- User defines when information is shown using <*Transparency numbers*>

If you want information to show up immediately: `<1->`

If you want information to show up only in the third set: `<3>`

Why and How Overlays Are Used

- Much like the transitions in PowerPoint
- Allows different information to be shown at different times on same slide
- User defines when information is shown using <*Transparency numbers*>

  If you want information to show up immediately: `<1->`

  If you want information to show up only in the third set: `<3>`

  If you want information to show up only in the second and fourth sets: `<2,4>`

The Overlay Feature

For Lists

```
\begin{enumerate}
  \item<1-> First item;
  \item<2-> Second item;\\
  ...
  \item<3-> Last item.
\end{enumerate}
```

The Overlay Feature

For Lists

```
\begin{enumerate}
  \item<1-> First item;
  \item<2-> Second item;\\
  ...
  \item<3-> Last item.
\end{enumerate}
```

1. First item;

The Overlay Feature

For Lists

```
\begin{enumerate}
  \item<1-> First item;
  \item<2-> Second item;\\
  ...
  \item<3-> Last item.
\end{enumerate}
```

1. First item;
2. Second item;

   ...

The Overlay Feature

For Lists

```
\begin{enumerate}
  \item<1-> First item;
  \item<2-> Second item;\\
  ...
  \item<3-> Last item.
\end{enumerate}
```

1. First item;
2. Second item;

   ...
3. Last item.

The Overlay Feature

For Non-Lists

You must use `\uncover<`*Transparency numbers*`>`

An Algorithm For Finding Primes Numbers.

```
\uncover<1->{int main (void)} \uncover<1->{\{}
\uncover<1->{std::vector<bool> is_prime (100,
true);} \uncover<1->{ for (int i = 2; i < 100; i++)}
\uncover<2->{{ if (is_prime[i])}} \uncover<2->{
\{} \uncover<3->{ std::cout << i << " ";}
\uncover<3->{ for (int j = i; j < 100;}
\uncover<3->{ is_prime [j] = false, j+=i);}
\uncover<2->{ \}} \uncover<1->{ return 0;}
\uncover<1->{\}}
```

```
int main (void) {
std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)




    return 0;
}
```

```
int main (void) {
std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)
 if (is_prime[i])
{


 }  return 0;
}
```

```
int main (void) {
std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)
 if (is_prime[i])
{  std::cout « i « " ";
 for (int j = i; j < 100;
 is_prime [j] = false, j+=i);
 }  return 0;
}
```

```
int main (void) {
std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)
 if (is_prime[i])
{  std::cout « i « " ";
 for (int j = i; j < 100;
 is_prime [j] = false, j+=i);
 }  return 0;
}
```

Using Verbatim

To use any sort of verbatim text, you must declare the frame as *fragile*:

```
\begin{frame}[fragile]
```

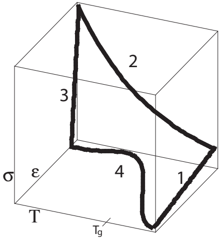
Ordering the Overlays

```
\begin{enumerate}
  \item \uncover<2,6>
  {Start at $T>T_g$ and deform}

  \item \uncover<3,6>
  {Cool below $T_g$}

  \item \uncover<4,6>
  {Release applied strain}

  \item \uncover<5,6>
  {Heat above $T_g$ and recovery}
\end{enumerate}
```



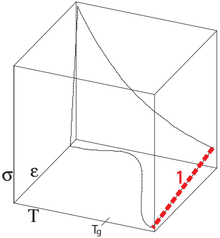
Ordering the Overlays

```
\begin{enumerate}
  \item \uncover<2,6>
  {Start at $T>T_g$ and deform}

  \item \uncover<3,6>
  {Cool below $T_g$}

  \item \uncover<4,6>
  {Release applied strain}

  \item \uncover<5,6>
  {Heat above $T_g$ and recovery}
\end{enumerate}
```



1. Start at $T > T_g$ and deform

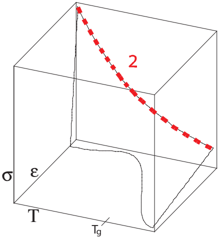
Ordering the Overlays

```
\begin{enumerate}
  \item \uncover<2,6>
  {Start at $T>T_g$ and deform}

  \item \uncover<3,6>
  {Cool below $T_g$}

  \item \uncover<4,6>
  {Release applied strain}

  \item \uncover<5,6>
  {Heat above $T_g$ and recovery}
\end{enumerate}
```



2. Cool below T_g

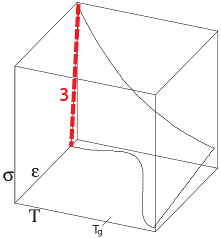
Ordering the Overlays

```
\begin{enumerate}
  \item \uncover<2,6>
  {Start at $T>T_g$ and deform}

  \item \uncover<3,6>
  {Cool below $T_g$}

  \item \uncover<4,6>
  {Release applied strain}

  \item \uncover<5,6>
  {Heat above $T_g$ and recovery}
\end{enumerate}
```



3. Release applied strain

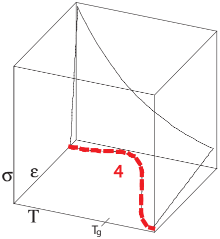
Ordering the Overlays

```
\begin{enumerate}
  \item \uncover<2,6>
  {Start at $T>T_g$ and deform}

  \item \uncover<3,6>
  {Cool below $T_g$}

  \item \uncover<4,6>
  {Release applied strain}

  \item \uncover<5,6>
  {Heat above $T_g$ and recovery}
\end{enumerate}
```



4. Heat above T_g and recovery

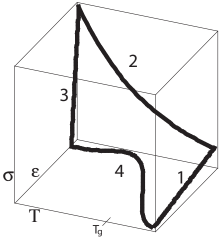
Ordering the Overlays

```
\begin{enumerate}
  \item \uncover<2,6>
  {Start at $T>T_g$ and deform}

  \item \uncover<3,6>
  {Cool below $T_g$}

  \item \uncover<4,6>
  {Release applied strain}

  \item \uncover<5,6>
  {Heat above $T_g$ and recovery}
\end{enumerate}
```



1. Start at $T > T_g$ and deform
2. Cool below T_g
3. Release applied strain
4. Heat above T_g and recovery

Overlaying Figures

In general, `\includegraphics<`*set(s) to show graphic*`}`
To overlay the figures on top of each other, use the command `\llap`

```
\llap{\includegraphics<1,6>[height=1.3in]{./figures/SMPThermoMechCycle}}%
\llap{\includegraphics<2>[height=1.3in]{./figures/ExpFig1}}%
\llap{\includegraphics<3>[height=1.3in]{./figures/ExpFig2}}%
\llap{\includegraphics<4>[height=1.3in]{./figures/ExpFig3}}%
\llap{\includegraphics<5>[height=1.3in]{./figures/ExpFig4}}
```

Inserting Figures

.eps or .ps files

Only when using `latex` and `dvips`

.pdf, .jpg, .jpeg or .png files

Only when using `pdflatex`

You Can Do Movies Too!

```
\usepackage{multimedia}
```

Thank You!

rdsiskin@ncsu.edu