

Impressive presentations with KeyJnote

SLIDE BALLET

KeyJnote is a PDF viewer with a difference, adding stylish 3D effects and powerful scripting to support impressive presentations.

BY MARTIN J. FIEDLER

The Linux environment offers several alternatives for creating presentations: MagicPoint, OpenOffice.org, Kpresenter, and LaTeX packages give users a rich choice of tools for creating attractive slides. However, these presentation tools often do not offer the full range of special effects that public speakers like to use to keep the discussion lively.

KeyJnote is a universal presentation tool for PDF files and images that provides some of the polish missing from the other tools in the Linux environment. KeyJnote gives users attractive transitions, helpful functions to focus audience attention, and an overview feature that makes it easier to navigate lengthy presentations.

Setting Up KeyJnote

KeyJnote was written in Python and uses OpenGL for display purposes. GhostScript handles rendering of PDF documents to provide display-ready image files, and the Pdftk tool provides excellent support for evaluating metadata. GhostScript and Python are both included with most modern Linux distribution.

You may need to download the various Python extensions from the Internet. KeyJnote requires PyOpenGL, PyGame, and PIL, the Python Imaging Library. Users with Debian-based distributions can resolve all of these dependencies in one fell swoop; Ubuntu users will need to enable the *Universe* repository before making the attempt:

```
# apt-get update
# apt-get install python z
python-opengl python-pygame z
python-imaging gs pdftk
```

You can download KeyJnote from SourceForge [1]. To run the program on Linux, you just need the 100KByte *.tar.gz* package. To install the program, become

Flexible

KeyJnote uses Info scripts that give users the ability to fine tune presentations. An Info script is a text file with the same name as the PDF file or image directory you want to present, but with an *.info* suffix. The Info script for *slides.pdf* would be *slides.pdf.info*, for example. Info files are actually Python scripts that can modify various presentation attributes.

One common application is that of controlling transition effects. For example, you can choose to use the spectacular page peel effect between the sections in a presentation, while keeping normal wipes between the pages. To do so, you can set the *transition* property of the *PageProps* variable individually for each page; *transition* specifies the transition effect between pages.

A typical Info script might look like the following:

```
AvailableTransitions.removez
(PagePeel)
PageProps = {
    4: { 'transition': PagePeel },
    9: { 'transition': PagePeel }
}
```

The first line removes the *PagePeel* effect from the list of *AvailableTransitions*, preventing KeyJnote from randomly assigning this effect to a page. The *PageProps* then stipulate the *PagePeel* effect for the transition between pages 4 and 5, and between pages 9 and 10. The KeyJnote help file describes various other options.



root, and copy the *keyjnote.py* file to */usr/local/bin*. You can omit the *.py* suffix to keep names consistent with the rest of the system:

```
$ tar xzf keyjnote-0.8.1.tar.gz
# cp keyjnote-0.8.1/keyjnote.py
  /usr/local/bin/keyjnote
```

Things are even easier for Gentoo users: to install the program in Gentoo, and resolve any dependencies, just enter *emerge keyjnote*.

You will need a computer with hardware-based 3D graphics acceleration if you expect KeyJnote to perform at its best. Although the program supports software rendering, this software rendering feature has a dramatic effect on the response time, and if you decide to use it, you will have to do without animations completely, which is a shame because animations are KeyJnote's biggest selling point.

Preparing a Presentation

As KeyJnote just basically displays slides, you can use some other program to create content. Software packages such as OpenOffice.org and LaTeX

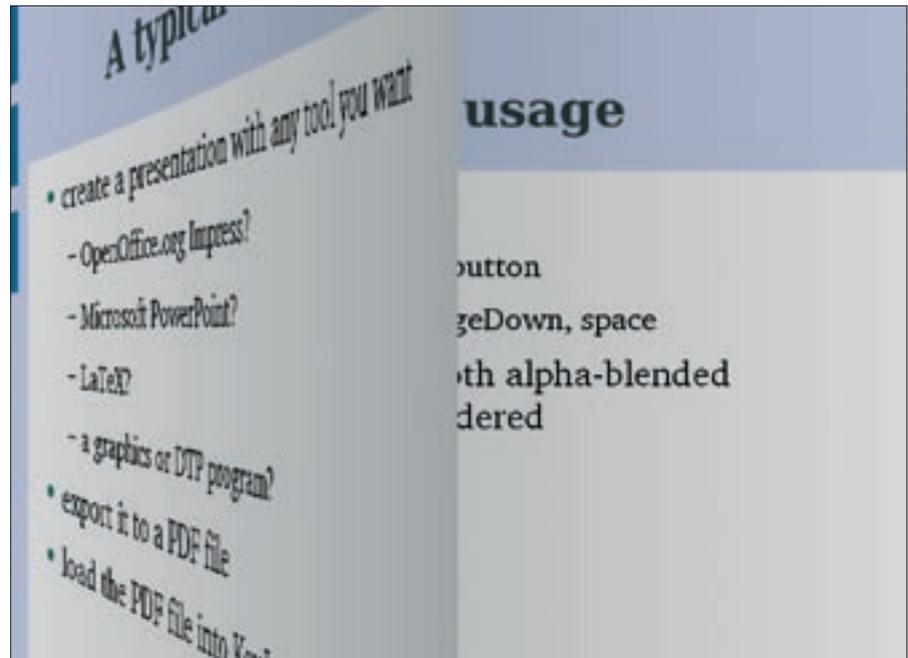


Figure 1: KeyJnote provides animated transition effects between pages.

are likely candidates. OpenOffice can export slides in the correct format via *File | Export as PDF...* If you use *pdflatex*, you may already be familiar with the popular *beamer* package, which also creates attractive PDF files.

If your choice of program can't create PDF files, or if the PDF export feature gives you some trouble, note that KeyJnote can handle PNG, TIF, or JPEG formatted images. Make sure the image files are stored below a directory without any subdirectories. The display order follows a simple alphanumeric pattern, which you will be familiar with from *ls*. The option of displaying images rather than PDFs means that KeyJnote can fulfill a second important task: a slide show with the photos from your last vacation will definitely benefit from KeyJnote's transition effects.

The Launch

After creating a PDF file – such as *slides.pdf* – launch KeyJnote by entering *keyjnote slides.pdf*. For a slideshow, just specify the name of the directory with the files (*not* the filenames) instead of the PDF filename. KeyJnote supports a number of command line switches, which you can list by setting the *-h* switch, and which are described in the manual that accompanies the program. The most important of these are *-f*, to switch to full-screen mode on launching the program, and *-g*, to set the screen

resolution. The following command line displays images from the *NewPhotos* directory in a 800x600 pixel window:

```
$ keyjnote -f -g 800x600
NewPhotos
```

When you launch KeyJnote, the program initially displays a splash logo and then a loading progress indicator. What the program is actually doing in the background is rendering the pages of the presentation in RAM to support quicker access later. KeyJnote also makes use of this launch time to populate the overview page.

You can click to interrupt this process if necessary. Setting the *-c* flag will disable the process, a useful option for machines that are low on memory resources, as it will also prevent KeyJnote

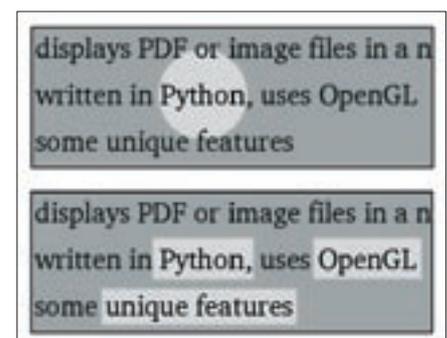


Figure 2: Various kinds of highlighting draw the audience's attention to the important parts of a page.

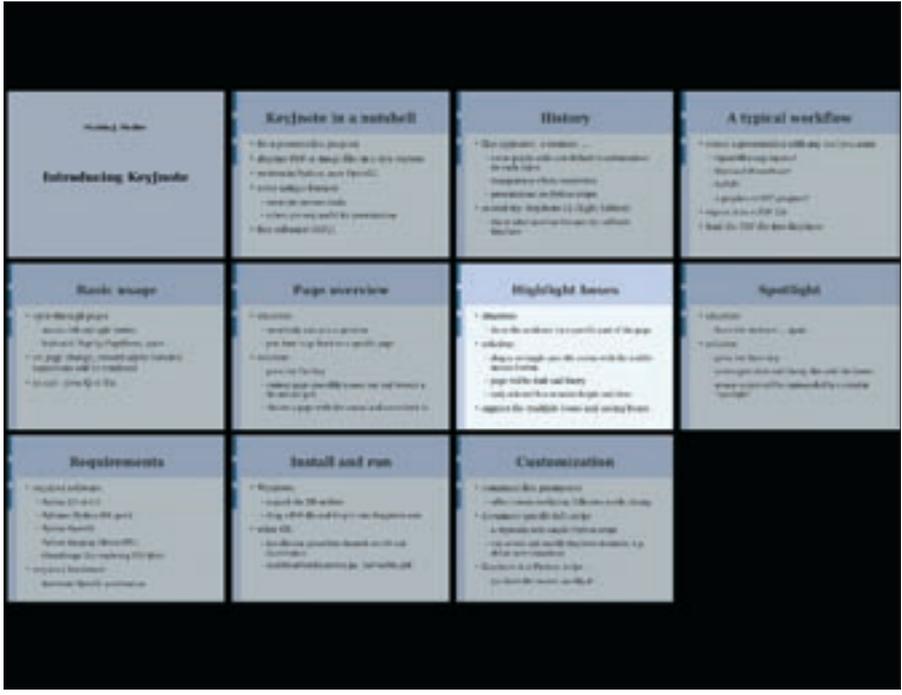


Figure 3: The overview page helps you find your way around longer presentations.

from storing previously displayed pages in RAM later.

Let the Show Begin

After displaying the startup and loading screens, KeyJnote jumps straight to the first page of the presentation. The program offers various methods of changing pages: the left mouse button [Pg-Dn] or the space key take you to the next page; the right mouse button [Pg-Up] or the backspace key take you to the previous page.

KeyJnote uses smooth transition effects when changing pages (Figure 1) and will normally choose the effect at random. Most of the transition effects are wipes, which you may be familiar with from Star Wars movies. But KeyJnote also has a neat page peel effect.

Effects such as smooth transitions are pure eye candy and don't have any genuine practical value. But KeyJnote's highlighting functions are a different matter. To draw the attention of the audi-

ence to important parts of the current slide, you can just press Enter. This turns on a kind of "spotlight" around the mouse cursor (Figure 2, top). While the rest of the slide becomes darker and goes slightly out of focus, the area around the cursor is illuminated to emphasize the importance of the information.

You can press the [+] and [-] keys to modify the spot radius, and pressing the Enter key a second time toggles the spotlight back off.

If you are not happy with a round spot, you can hold down the left mouse button and drag the mouse to draw a highlighting frame. Multiple frames per page are possible (Figure 2, bottom). To remove highlighting, just right click a frame.

From time to time, you might need to zoom in on a slide to display smaller sketches in more detail. If you selected PDF as the input format, you can press the [Z] key to toggle 200% zoom on and off. In KeyJnote's zoom mode, you can hold down the right mouse button and drag the mouse to pan the visible display.

The overview page (Figure 3) is another useful feature of KeyJnote. You can press the Tab key to zoom out of the current slide and display a tiled preview of all the slides in the presentation. You can then use the mouse to select the slide you need and click to return to the full-size display.

Troubleshooting

Unfortunately, KeyJnote was hit by a few bugs in our testing. The most common causes of error were OpenGL and GhostScript.

KeyJnote always keeps the images of the current and next pages in an OpenGL texture. As textures need dimensions that are powers of two, KeyJnote generally rounds up the size to 1024x1024 pixels for a working area of 1024x768 pixels or 2048x1024 for 1280x1024 pixels. Unfortunately, not all OpenGL implementations can handle textures of this size. In other cases, the graphics memory is simply not big enough. In a best case scenario, the zoom feature would just not work, and in the worst case, you would need to drop the screen resolution down to 1024x768 using the `-g` command line option. If this doesn't help,

you have a major problem: the next smallest resolution that really saves memory space is 512x384.

The second most common cause of errors are PDF files that conflict with the installed GhostScript version. GhostScript 7.x will not display files created using *pdf_latex* for example, and it gets a few characters completely wrong. In other PDFs font anti-aliasing may not work, or it may be unsatisfactory. However, the CPU-intensive `-s` option might give you a workaround here. In many cases, especially with OpenOffice.org exports, GhostScript will issue warnings about font errors, although the display is fine. As a general rule, the more recent your version of GhostScript is, the fewer problems you are likely to encounter.

Conclusions

KeyJnote is a small but effective presentation tool that will soon have the audience on your side. Unfortunately, the program is quite exacting in its hardware requirements. The memory requirement for pre-rendering in KeyJnote is enormous – 2.5MB per slide at a resolution of 1024x768 pixels.

The delays on launching the program can be a problem with lengthy presentations. The upcoming version 0.9 promises to resolve these issues by pre-rendering the presentation in the background and storing the results temporarily on disk. ■

INFO

[1] KeyJnote project homepage:
<http://keyjnote.sourceforge.net/>