The PDFAnim_temp package

Jochen Skupin (Version 0.52A, modified by John Bowman)

December 29, 2006

Abstract

A LaTeX package to create animated PDFs with pdfLaTeX.

Contents

1 Introduction

PDFAnim_temp is an interim LaTeX package to create animated PDFs with pdfLaTeX. It will become obsolete as soon as the 0.53 release of PDFanim is made.

As it is very unusual to create animations with LaTeXand to have to deal with JavaScript it might be that large parts of this document are not understandable for the normal LaTeXuser. Don't be too frustrated about it and just try the examples.

Comments and bug reports are *very* welcome. You can contact me at:

jochen.skupin@uni-bremen.de

Don't be too disappointed if it takes a while until you get an answer, as I have little spare time to work on this package.

2 Usage

To include an animation in a PDF file you need:

- PDFLaTeX
- the PDFAnim package (i.e. the `pdfanim_temp.sty`-file)
- The frames of the animation as single PDF files with filenames in the form `filename`n`.pdf` where n is the number of the nth frame of the animation starting with 0.
- Adobe Acrobat or AcroReader to view the animation

The package includes the single frames as hidden picture buttons in the PDF-file. The animation itself is created using a visible picture button whose picture is exchanged using the JavaScript language available in Adobe Acrobat and AcroReader. The frames of the animation have to be included in the document preamble using

```
\PDFAnimLoad[options]{name}{filename}{number_of_frames}
```

where `filename` is the head of the filename of the PDF-files containing the single frames of the animation omitting the running number and the `.pdf` extension. `name` is the name of the animation. With this name it can be accessed in the document. `number_of_frames` gives the number of files to be included. An animation with 10 frames would require `number_of_frames` = 10 and files `filename0...9.pdf`. The various `options` of `\PDFAnimLoad` will be discussed in section 4.
To access the animation anywhere in the document use:

```
\PDFAnimation{name}
```

Examples can be found at:

- `http://www.uni-bremen.de/~skupin/pdfanim/Demos`
- `http://www.tug.org/texshowcase/`

3 Package option(s)

The package has to be loaded in the document preamble with:

```
\usepackage[options]{pdfanim_temp}
```

The package `options` change the way the JavaScript code is included in the PDF-file.

`default` The default behaviour (by giving no options at all) is to include most of the JavaScript on document level and access it via page open/close attributes or via the on-click action of the picture button.

- The advantage is that only very short function calls have to be included in the page open/close attributes because most JavaScript code is stored on document level. Also it is possible to define global variables to store the status of an animation (like the last displayed frame, is it running or not).
-

`NoDocJS` With this option no document level JavaScript is generated. All JavaScript is stored in the picture button itself and the page open/close attributes.

- The disadvantage is that all JavaScript code is stored on every page again. This leads to slightly bigger file sizes. Due to this all variables are newly initialised on every page so it is not possible to have global variables to remember the status of an animation from page to page.

`NoPageJS` I didn't yet manage to derive on which page a distinct animation (especially floats) is placed by TEX. The JavaScript code for all animations is therefore included in every page. Not very elegant—I know. The problem is that when the document contains auto starting animations all of them are started again on *every* page. With the `NoPageJS` option the automatic inclusion of JavaScript code on any page can be suppressed.

- The advantage is that not all auto start animations will run permanently.
- The disadvantage is that the user has to take care to put the JavaScript on pages with animations by hand. To do this the following 4 commands are provided:

`\PDFAnimJSPageEnable` enable JavaScript in page-attribute of current page

`\PDFAnimJSEnable` enable JavaScript in page-attribute on all following pages

`\PDFAnimJSPageDisable` disable JavaScript in page-attribute of current page

`\PDFAnimJSDisable` disable JavaScript in page-attribute on all following pages

4 PDFAnimLoad options

There are already much too many options. But during my usage of PDFAnim I needed all of them for one or the other reason. Unfortunately not all options work well together and some of the older ones might not work at all in the current implementation anymore. Maybe some clean-up is needed here.

4.1 Boolean options

`single` extract frames as the individual pages of a single input file

`auto` auto start animation

`debug` enable debug messages

`fallback` include start-picture below PictureButton as fallback solution for xpdf, macs ... (gives poor animation results)

`hidden` create hidden PictureButton (mostly for internal use)

`loop` loop animation

`noclick` don't recognise clicks on PictureButton

`remember` remember last displayed picture when changing to another page

`reverse` play animation in reversed order

`step` advance animation on every mouse-click

4.2 Options that take parameters

`bcolor` bordercolor of PictureButton

`bgcolor` backgroundcolor of PictureButton

`defaultframe` select frame to display when animation not yet running

`depth` depth of PictureButton

`extension` set extension of included pictures (till now only pdf works)

`height` height of PictureButton

`interval` set interval in ms between animation frames (only shows effect if interval is longer than the time needed to display a frame)

`name` name PictureButton (may be useful for further editing of the pdf, not needed/used by PDFAnim)

`onclick` JavaScript action to perform on mouse-click (mostly for internal use)

`scale` scaling of picture used in PictureButton:

- A Always scale.
- B Scale only when the icon is bigger than the annotation rectangle.
- S Scale only when the icon is smaller than the annotation rectangle.
- N Never scale.

`scaletype` type of scaling of picture used in PictureButton:

A Anamorphic scaling: scale the icon to the annotation rectangle exactly, without regard to its original aspect ratio (ratio of width to height).

P Proportional scaling: scale the icon to the width or height of the annotation rectangle while maintaining the icon's original aspect ratio. If the required horizontal and vertical scaling factors are different, use the smaller of the two, centering the icon within the annotation rectangle in the other dimension.

`use` use pictures from other animation (to save memory)

`usecnt` use counter from other animation

`startframe` select first frame to display

`width` width of PictureButton

4.3 Navigation buttons

Another command, `\PDFAnimButtons`, which is available if you also load D. P. Story's package `eforms.sty`, presents an alternative method for starting and stoping animations by pressing the Play/Pause button. It also provides a method for jumping to arbitrary locations in long animations. I the animation is not running, you can input a percentage; pressing Play/Pause then starts the animation from that relative position (e.g. 75 means frame 150 for a 200-frame animation).

5 Open issues

Fewer options

During my usage of PDFAnim I used all of the options in section 4. But I guess they can be reduced and combined in a better way. Any improvements are welcome.