

# The interactiveanimation Package

Luis González, Javier Toro, Pedro Linares

June 6, 2013

## Abstract

A  $\LaTeX$  package for creating branching and button controllable PDF animations from image files.

*Keywords:* build button controllable animation branching animation  $\LaTeX$  pdf $\LaTeX$  JavaScript Adobe Reader

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>1</b>
<b>3</b>	<b>Installation</b>	<b>2</b>
<b>4</b>	<b>Using the package</b>	<b>2</b>
<b>5</b>	<b>The user interface</b>	<b>2</b>
5.1	Inserting the animation . . . . .	2
5.2	Inserting frames . . . . .	3
5.3	Inserting control buttons . . . . .	3
5.3.1	Control button options . . . . .	3
<b>6</b>	<b>Examples</b>	<b>5</b>
<b>7</b>	<b>Known issues</b>	<b>6</b>

## 1 Introduction

This package provides an interface to create Portable Document Format (PDF) files with branching animations. Animations are built up as a set

of concatenated interactive frames. Each interactive frame can be set to display an image and a set of buttons that allows the course of the overall animation to be controlled by the user. Buttons can be configured to transfer the action focus to another interactive frame and to display an animated image sequence before the forthcoming frame is displayed.

By now, the only supported way for compiling using this package is by producing the PDF file with pdfTeX.

The final PDF can be viewed in current Adobe Readers on all supported platforms as long as they have JavaScript support.

## 2 Requirements

pdfTeX, version  $\geq 1.20$

Adobe Reader, version  $\geq 6$

## 3 Installation

Unzip the file ‘interactiveanimation.zip’ and copy the ‘interactiveanimation.sty’ and the folder ‘javascript’ in a place where L<sup>A</sup>T<sub>E</sub>X can find them.

MiK<sub>T</sub>TeX users should run the package manager for installation.

## 4 Using the package

Invoke the package by putting the line

```
\usepackage{interactiveanimation}
```

to the preamble of your document, i. e. somewhere between `\documentclass` and `\begin{document}`.

## 5 The user interface

Interactive animations are defined using three basic instructions provided by this package:

- The animation is inserted, using the environment `animation`.
- Frames are inserted, using the environment `aframe` inside of `animation`.

- Control buttons are inserted, using the command `\controlbutton` inside `aframe`.

```

\begin{animation}{<width>}{<height>}
\begin{aframe}{<frame name>}{<image file>[,<file page>]}
\controlbutton[<sequence file>,<start page>[,<end page>]]
{<caption>}{<target frame>}{<options>}
    ... other buttons
\end{aframe}
    ... other frames
\end{animation}

```

## 5.1 Inserting the animation

The `<width>` and `<height>` arguments must be given in valid  $\TeX$  dimensions. Horizontal and vertical dimensions of the animation are scaled independently to fit `<width>` and `<height>`.

## 5.2 Inserting frames

If present, argument `<frame name>` is used in control buttons to identify which frame will be displayed when the button is pressed, so it should be unique in current animation.

`<image file>` and `<file page>` specify a page of a PDF file to be displayed as an image by the frame. If `<file page>` is not specified, first page is assumed; if any image is not specified, a blank frame will be displayed.

## 5.3 Inserting control buttons

A control button may display, before changing the current frame, a sequence of images. All of those images must be into the same external PDF file. If you want to display a sequence, you should specify the file where images are stored in `<sequence file>` and the start and end page range in `<start page>` and `<end page>`, respectively; if only `<start page>` is given in the range, first page is assumed as range start and the given number as range end. Range may also be decreasing if `<end page>` is less than `<start page>`.

`<caption>` gives a text to be displayed by the control button.

`<target frame>` specifies the name of a frame to be displayed when that control button is pressed, you may also, instead of specifying a frame name, use `first`, `last`, `previous` or `next` for first, last, previous and next frame

defined into the animation, respectively. If not specified, current frame is assumed.

`<options>` a comma-separated list that describes the appearance and behavior of the control button. See section below for more details.

### 5.3.1 Control button options

`X=<coordinate> | start | center | end`

`Y=<coordinate> | start | center | end`

give `X` and `Y` coordinates of a point to take reference for specifying where button's drawing should start (see below). The coordinates should be non-negative integers and represent a point into the animation. Point `X=0,Y=0` will be the lower left corner.

Instead of given a number, you can also use `start`, `center` and `end` to specify the start, center or end (`X` or `Y`) coordinates of the animation, respectively. For example, for an animation of  $400 \times 300$ , `X=end` would be equivalent to `X=400` and `Y=center` would be equivalent to `Y=150`.

Note that `X` and `Y` are uppercase, so if you use lowercase it will not work.

`position=left | right | above | below | center | from`

Describes where the button will be positioned relative to the given point. For example, if `position=right` is specified, the button will be at the right of the given point, if `below`, it will be below of the point and so on. If `from` is specified, the point will be the upper left corner of the button.

Note: if `position` is not specified, the button will be at upper corner of the animation for first button or at right for other buttons. If it is specified but it was not given any point, the reference will be the last inserted button.

`width=<width>`

`height=<height>`

`scale=<factor>`

By default, button's width and height are automatically set to fit button's caption. However, you may also set specific dimensions using those options. If specified, `width` and `height`'s argument must be a non-negative integer representing a distance in points. In addition, instead of setting manually the dimensions, you may scale automatically fixed dimensions using `scale`; that allows you to increase the size of a button without breaking the proportion; for example, `scale=2` will multiply width and height by 2.

`hidden`

Makes the button becomes visible only when the mouse is over it and invisible otherwise.

`transparent`

Makes visible only button's caption, the button still being pressable. If you want a fully transparent button, you just should specify a blank caption.

Note: if you want to define a transparent button on the same area of others, make sure to define it before those other buttons so it is not over them to prevent from activating.

`span=<milliseconds>`

If the button has an associated sequence of images, this option sets the time span, in milliseconds, for images change. Default value is 1000.

`keep`

Keep running the image sequence until frame is pressed.

## 6 Examples

The following example is an analog clock and consist of four frames. Each frame displays an interval of 15 minutes on the clock and has two buttons to going back and going forward. Before going back and forward, the buttons display a sequence which represents the movement that performs hand of the clock to reach, passing throught every minute between original and destination positions.

The file 'clock.pdf', where images are stored, consists of 60 images corresponding with 60 minutes of an our.

```
\begin{animation}{450 pt}{450 pt}
\begin{aframe}{12:00}{clock.pdf,60}
\controlbutton[clock.pdf,60,45]{<< Backward}{12:45}{%
position=right, X=start, Y=start, span=75}
\controlbutton[clock.pdf,1,15]{Forward >>}{12:15}{%
position=left, X=end, Y=start, span=75}
\end{aframe}

\begin{aframe}{12:15}{clock.pdf,15}
\controlbutton[clock.pdf,15,1]{<< Backward}{12:00}{%
```

```

position=right, X=start, Y=start, span=75}
\controlbutton[clock.pdf,15,30]{Forward >>}{12:30}{%
position=left, X=end, Y=start, span=75}
\end{aframe}

\begin{aframe}{12:30}{clock.pdf,30}
\controlbutton[clock.pdf,30,15]{<< Backward}{12:15}{%
position=right, X=start, Y=start, span=75}
\controlbutton[clock.pdf,30,45]{Forward >>}{12:45}{%
position=left, X=end, Y=start, span=75}
\end{aframe}

\begin{aframe}{12:45}{clock.pdf,45}
\controlbutton[clock.pdf,45,30]{<< Backward}{12:30}{%
position=right, X=start, Y=start, span=75}
\controlbutton[clock.pdf,45,60]{Forward >>}{12:00}{%
position=left, X=end, Y=start, span=75}
\end{aframe}
\end{animation}

```

## 7 Known issues

When trying to insert an animation with not enough space in current page, the animation will be placed in the next one. However, only the part which could not fit in previous page will be visible. The simplest solution to solve it is inserting a page skip before the animation, e.g. `\newpage`.