The aobs-tikz package*

Claudio Fiandrino
claudio.fiandrino@gmail.com

January 12, 2014

Abstract

The package provides auxiliary styles helpful for drawing overlayed pictures in Beamer. These styles should be intended as extension of the previous work shown by user Daniel of TeX.SX in Mindmap tikzpicture in beamer (reveal step by step).

Contents

1 Introduction

The aim of aobs-tikz is to provide users simple tools to create overlayed-aware pictures for Beamer presentations. A set of new TikZ styles, grouped within a library, has been define on purpose.

The library is proposed as extension of the previous work by user Daniel on TeX.SX, illustrated in Mindmap tikzpicture in beamer (reveal step by step). The very first version of the library has been implemented for answering the question Highlighting in Beamer using TikZ nodes: aobs-tikz further extends and improves the original styles.

The main advantage of using the new styles is that they automatically prevent the so called *jumping-effect* which occasionally happen with Beamer.

*This document corresponds to aobs-tikz v1.0, dated January 12, 2014. It is released under and subject to the latest version of the LaTeX Project Public License (LPPL).

2 The new styles

The new styles can create overlays by altering the colors and the aspect of pictures' elements. Usually, to create an animation, the elements have to appear or disapper, the shading, the fill or the border color should change. To achieve this goal, three features have to be specified: the normal behavior, the modified behaviour and the moments in which the modified behaviour takes place. aobs-tikz defines TikZ styles for these three features accordingly . For example, to alter the shape filling, it is possible to exploit:

- `background default fill=<style>`: the style used for default behaviour;
- `background fill=<style>`: the style used for the modified behaviour;
- `fill on=<overlay specifications>`: moments in which the modified behaviour is activated.

The new styles can alter the following TikZ properties:

- *fill*;
- *draw*;
- *filldraw*;
- *text*;
- path aspect, including thickness, double line and pattern (solid, dashed, dotted, etc. . .);
- *shade*;
- *shadedraw*.

`draw` To alter the border color properties, the following options are available:

- `background default draw=<style>`;
- `background draw=<style>`;
- `draw on=<overlay specifications>`.

`filldraw` To alter the both filling and border color properties, the following options are available:

- `background default filldraw=<border-col filled by fill-col>`;
- `background filldraw=<border-col filled by fill-col>`;
- `filldraw on=<overlay specifications>`.

`text` To alter the `text` color properties, the following options are available:

- `background default text=<style>`;
- `background text=<style>`;
- `text on=<overlay specifications>`.

`path aspect` To alter the path aspect, the following options are available:

- `background default aspect=<style>`;
- `background aspect=<style>`;
- `aspect on=<overlay specifications>`.

`shade` To alter the shading properties, the following options are available:

- `background default shade=<style>`;
- `background shade=<style>`;
- `shade on=<overlay specifications>`.

`shadedraw` To alter both filling and border color properties, the following options are available:

- `background default shadedraw=<style>`;
- `background shadedraw=<style>`;
- `shadedraw on=<overlay specifications>`.

3 Implementation

3.1 Package

The package itself loads only TikZ and the library `overlay-beamer-styles`.

```
\RequirePackage{tikz}
\usetikzlibrary{overlay-beamer-styles}
```

3.2 TikZ Library

The core of the package is the TikZ library `overlay-beamer-styles`. The first style defined is `visible on` based on prior work by user Daniel of TeX.SX in Mindmap tikzpicture in beamer (reveal step by step). The original style has been enforced to make it working also in presence of opaque text.

```
\tikzset{
    invisible/.style={opacity=0,text opacity=0},
    visible on/.style={alt=#1{}{invisible}},
    alt/.code args={<#1>#2#3}{%
      \alt<#1>{\pgfkeysalso{#2}}{\pgfkeysalso{#3}}
```

```
    },
}

\tikzset{
  background text/.style={text=#1},
  background text/.default={black},
  background default text/.style={
    background text/.default={#1},
  },
  text on/.style={alt=#1{}{background text}},
}

\tikzset{
  background fill/.style={fill=#1},
  background fill/.default={white},
  background default fill/.style={
    background fill/.default={#1},
  },
  fill on/.style={alt=#1{}{background fill}},
}

\tikzset{
  background draw/.style={draw=#1},
  background draw/.default={white},
  background default draw/.style={
    background draw/.default={#1},
  },
  draw on/.style={alt=#1{}{background draw}}
}

\tikzset{
  background filldraw/.style args={#1 filled by #2}{draw=#1, fill=#2},
  background filldraw/.default=white filled by white,
  background default filldraw/.style={
    background filldraw/.default={#1},
  },
  filldraw on/.style={alt=#1{}{background filldraw}},
}

\tikzset{
  background aspect/.style={#1},
  background aspect/.default={white},
  background default aspect/.style={
    background aspect/.default={#1},
  },
  aspect on/.style={alt=#1{}{background aspect}},
}

\tikzset{
  background shade/.style={#1},
```

```
  background shade/.default={top color=white, bottom color=white},
  background default shade/.style={
    background shade/.default={#1},
  },
  shade on/.style={alt=#1{}{background shade}},
}

\tikzset{
  background shadedraw/.style 2 args={draw=#1, #2},
  background shadedraw/.default={white}{top color=white, bottom color=white},
  background default shadedraw/.style={
    background shadedraw/.default={#1},
  },
  shadedraw on/.style={alt=#1{}{background shadedraw}},
}
```

At this point, some comments are needed on the subsequent option `double disabled`. For the best of my knowledge, this is option is not implemented in PGF 2.10 either in PGF 3.0.0, but it is absolutely relevant to the scope of this library. Suppose you wish to alter a double path by removing in some moments its double property: without the following option, it would not be possible (at least without redrawing the path).

```
% option for disabling double when not needed in
% subsequent overlays
\tikzoption{double disabled}[0pt]{%
  \pgfmathsetlength{\pgf@x}{#1}%
  \edef\tikz@double@setup{%
    \pgf@x=\the\pgf@x%
    \pgflinewidth=\pgf@x%
    \noexpand\pgfsetlinewidth{\pgflinewidth}%
    \noexpand\pgfsetinnerlinewidth{\the\pgf@x}%
  }%
\tikzset{double}}
```

4 Example

It follows a complete example which exploits all the defined styles. The first frame mainly reports showcases of border, filling and shading properties modifications. The second frame shows examples of modifications for the remaining properties, including text color and path aspect.

```
\documentclass{beamer}
\usepackage{lmodern}
\usepackage{tikz}
\usetikzlibrary{positioning,
  shapes.geometric,
  shadows
}
% loading new library
```

```
\usetikzlibrary{overlay-beamer-styles}

\definecolor{processblue}{cmyk}{0.96,0,0,0}

\begin{document}
\begin{frame}{Styles for draw, fill and shading modifications}
\begin{columns}[T]
\begin{column}{0.2\textwidth}
\centering
Fill draw\\[2ex]
\tikz[baseline=(A.base)]{%
\node[background fill=red!50,%
      fill on=<2>,%
      anchor=base,%
      rounded corners,%
      ] (A) {ABCD};
}

\tikz[baseline=(A.base)]{%
\node[background fill=blue!50,%
      fill on=<{1,3}>,%
      anchor=base,%
      rounded corners,%
      ] (A) {EFGH};
}

\tikz[baseline=(A.base)]{%
\node[background draw=red,%
      draw on=<2>,%
      anchor=base,%
      rounded corners,%
      ] (A) {IJKL};
}

\tikz[baseline=(A.base)]{%
\node[background draw=blue,%
      draw on=<{1,3}>,%
      anchor=base,%
      rounded corners,%
      ] (A) {MNOP};
}

\tikz[baseline=(A.base)]{%
\node[background filldraw=red filled by blue!10,%
      filldraw on=<2>,anchor=base,%
      rounded corners,%
      ] (A) {QRST};
}
\end{column}
\begin{column}{0.2\textwidth}
```

```
\centering
Shadings\\[2ex]
\tikz[baseline=(A.base)]{%
\node[background shade={top color=red!50, bottom color=white},%
      shade on=<2>,%
      anchor=base,%
      rounded corners,%
      ] (A) {ABCD};
}

\tikz[baseline=(A.base)]{%
\node[background shade={inner color=red!50, outer color=white},%
      shade on=<{1,3}>,%
      anchor=base,%
      rounded corners,%
      ] (A) {EFGH};
}

\tikz[baseline=(A.base)]{
\node[background shade={left color=orange!50, right color=white},%
      shade on=<2>,%
      anchor=base,%
      rounded corners,%
      ] (A) {IJKL};
}

\tikz[baseline=(A.base)]{
\node[background shadedraw={blue}{top color=white, bottom color=cyan!30},%
      shadedraw on=<{1,3}>,%
      anchor=base,%
      rounded corners,%
      ] (A) {MNOP};
}

\tikz[baseline=(A.base)]{
\node[background shadedraw={green!50!black}{inner color=white,%
      outer color=green!30},%
      shadedraw on=<2>,%
      anchor=base,%
      rounded corners,%
      ] (A) {QRST};
}
\end{column}
\begin{column}{0.55\textwidth}
\centering
Node application\\[2ex]
\begin{tikzpicture}[node distance=0.5cm]
```

Of course, it is always possible to group in high-level styles the styles provided by aobs-tikz.

```
\tikzset{visibility 1/.style={
      background draw=red, draw on=<{1,4}>,
      background shade={top color=white,
      bottom color=red!30},
      shade on=<{2,3}>,
      }
}
\tikzset{visibility 2/.style={
      background shadedraw={green!50!black}{inner color=white,
      outer color=green!30},
      shadedraw on=<{2,3}>,
      }
}
\tikzset{visibility 3/.style={
      background draw=orange,
      draw on=<1->,
      background fill={orange!30},
      fill on=<{2,3}>,
    }
}
```

The following high-level style shows that the new styles can be combined to obtained more fine results. Specifically, rather than using a *shadedraw* modification, the `visibility 4` style exploits separately *shade* and *draw* modifications to having them visible in different overlays.

```
\tikzset{visibility 4/.style={
      background draw=purple,
      draw on=<2->,
      background shade={left color=purple!30,
      right color=cyan!30},
      shade on=<{3,4}>,
    }
}
\node[trapezium,
      visibility 1] (A) {Text};
\node[trapezium,
      visibility 2,
      below= of A] (B) {Text};
\node[trapezium,
      visibility 3,
      below= of B] (C) {Text};
\node[trapezium,
      visibility 4,
      below= of C] (D) {Text};
\end{tikzpicture}
\end{column}
\end{columns}
\end{frame}

\begin{frame}{Styles for path aspect and text color modifications}
```

```
\centering
\begin{tikzpicture}[node distance=3cm and 2cm,
      semithick ,
      state/.style={circle,
            top color=white,
            bottom color=processblue!20,
            draw, processblue,
            text=blue,
            minimum width=1cm},
      background default shade={top color=white,
            bottom color=processblue!20},
      background default draw={processblue,
            semithick}]
```

Shadows can be managed with the help of the style `visible on`: it follows an example with a *circular drop shadow*.

```
\node[state,
      background draw={blue!80,
            line width=1mm},
      draw on=<2>,
      circular drop shadow={visible on=<2>},
      visible on=<{1,2}>% NOT visible in 3
     ] (C) {$1$};
\node[state,
      background draw={orange},
      draw on=<{1,3}>,
      background default aspect={semithick,
            double disabled},
      background aspect={double},
      aspect on=<{1,3}>,
      background shade={top color=white,
            bottom color=orange!30},
      shade on=<{1,3}>,
      above left= of  C] (A) {$0$};

\node[state,
      background text=violet,
      background default text=red,
      text on=<2>,
      above right= of C] (B) {$2$};

\draw (A)-- (B) (C)-- (A);

\draw[background default aspect={solid,semithick},
      background aspect={dashdotted,
            very thick},
      aspect on=<{2,3}>,
      background default draw={black},
      background draw={red},
      draw on=<3>](B)-- (C);
```

```
\end{tikzpicture}
\end{frame}
\end{document}
```