Developer's Guide on
SJTUBeamer MIN

Log Creative

0.9-LTS June 6, 2021

Contents

1   Preface

SJTUBeamer ⸤MIN⸥ is a presentation template based on beamer package in LaTeX, to fulfill the enthusiasm of those SJTU users to present their content nicely, benefiting from the technology of TeX typesetting engine.

This is a Developer's Guide on SJTUBeamer ⸤MIN⸥ . The document is written in English because the operation in this guidance could be dangerous. Be careful when playing with those macros.

SJTUBeamer ⸤MIN⸥ — the minimal work set of SJTU VI

⸤MIN⸥ - *minimal*:	minimal work set of SJTU VI.
⸤MIN⸥ - *minimalism*:	designed in the style of minimalism.
⸤MIN⸥ - *minimum*:	minimum shapes to show your content.

2   Compliation

Most problems come from LaTeX compilation. The required packages are in the following list.

pgfplots	tikz	xcolor
pgfplotstable	sansmath	tcolorbox
ctex	biblatex	beamer

The detailed description is documented below.

2.1   MiKTeX

All required packages will be automatically installed if you are using MiKTeX[1]. And if you want to use the `latexmk` command, please install Perl[2] first. And the compilation command for SJTUBeamer ⸤MIN⸥ is as follows:

```
latexmk -pdf main -interaction=nonstopmode
```

2.2   TeX Live

Since some packages are not default installed in the full release of TeX Live, you have to install the packages manually.

On Ubuntu, you could install `pgf` and `xcolor` and other drawing packages through the following command[3]:

```
sudo apt install texlive-pictures
```

To typeset Chinese characters, you would better use `CJKutf8` package (in SJTUBeamer ⌷MIN⌷, set `[cjk=true]`), since it is compatible with all platforms and multiple language support. Surround `CJK` environment to make it work and remember to move all the Unicode characters in the permeable to the `CJK` environment[4]:

```
\begin{document}
\begin{CJK}{UTF8}{gbsn}
    \institute[]{}
    \title{}
    \subtitle{}
    \author{}
    \date{}
    % your content here ...
\end{CJK}
\end{document}
```

However, if you stick into `ctex`, you can install through `tlmgr`. If that works, then we call it a day.

```
sudo tlmgr install ctex
```

Sometimes, you installed an old TeX Live, and you have to upgrade the `tlmgr` for the new version. And the process could be very buggy, since the following warning may be shown:

```
unexpected return value from verify_checksum: -5
```

and to upgrade the `tlmgr` is painful on Ubuntu. You should add the following content to `/etc/profile/`, which will add the newest path when the system is booting up[5]:

```
export PATH=/usr/local/texlive/2021/bin/x86_64-linux:
/usr/local/texlive/:$PATH
```

Reboot your computer if necessary. Then the compile system will be moved to the newer version of TeX Live. Try to install the corresponding packages through the GUI interface of `tlmgr`:

```
sudo tlmgr update --self
sudo tlmgr gui
```

And if you encountered that

```
Critical Package ctex Error: CTeX fontset 'fandol' is
unavailable in current(ctex) mode.
```

You have to modify your compiling program from pdfLATEX to XeLATEX by adding the following magic command on the first line:

```
% !TeX TS-program = xelatex
```

2.3 Boost Up

However, it has been tested that the compilation on SJTUBeamer MIN is slow. Since the complex patterns have to be rendered in vector shapes and the bibliography requires multiple times of compilation, the time could be wasted on some repetitive works.

This scenario could be improved by enable `[pattern=none]` option on SJTUBeamer MIN and enable `[draft]` option on beamer. The former one will disable all the pattern rendering, and the latter one will ignore all the TOC (table of contents) generating.

The project has been implanted to Overleaf. Here is the link [6]. And to make that works, the compilation on TEX Live 2021 has to be implemented. And it is discovered that setting the document information outside the `document` environment will cause a significantly longer compiling time, which may be caused by some improper settings in CTEX package. The workaround of that is to follow the setup mentioned in `CJK` settings: put that info into the body of document[4].

Currently, CI is available on Github Actions by compiling on LuaLATEX. SJTUBeamer MIN uses `xu-cheng/latex-action@v2` for the compilation docker [7] and relocates the compiling folder to `src/`. After compiling, output the PDF artifact. See `.github/workflows/main.yml` for details.

At the same time, AutoBeamer[8] is making its own effort on generating beamer code automatically by some replacing strategies. You could preview your beamer code through conversion on Markdown or the article LATEX code.

Furthermore, there is space for boosting up the beamer compilation time by making use of multi-core processors. Since it is a frame-based document, and the connection between each frame is loose (only some page numbers and citations need to be calculated), the multi-threaded compilation is possible for the beamer class. You can glimpse the multi-threaded processing for LATEX from the package `animate`. In fact, the author created some batch compiling work[9] together with the `-Parallel` parameter in PowerShell 7 to make full use of the concurrent computer architecture.

3 Modular Architecture

By the recommendation from beamer package[3], SJTUBeamer ⟨MIN⟩ uses the same modular architecture to build the template. Like it is in Java, to let the beamer template locate your theme, the style file has to be in the standard name.

.sty File	Description
beamercolorthemeSJTUBeamermin.sty	Define global color schemes.
beamerfontthemeSJTUBeamermin.sty	Set the font format.
beamerinnerthemeSJTUBeamermin.sty	Specifies all parts inside a frame.
beamerouterthemeSJTUBeamermin.sty	The frame header and bottom bar.
beamerthemeSJTUBeamermin.sty	Entry point of the theme.

Notice that there are some dependencies (logo files) in the `vi/`. Copying the `vi` folder is necessary. Or you could define the location of the logo file by giving `\logo{\includegraphics{logo.pdf}}`.

main.tex			
beamerthemeSJTUBeamermin.sty			
colortheme.sty	fonttheme.sty	outertheme.sty	innertheme.sty
		logo.pdf	

3.1 Theme

The main theme file `beamerthemeSJTUBeamermin.sty` is the entry point of the theme template. For users, after acquiring the beamer package, `\usetheme` command will serve as the caller of the theme.

```
\documentclass{beamer}
\mode<presentation>
\usetheme{SJTUBeamermin}
```

And this file will preprocess the option passed to the theme. Some options will be affected immediately, while others will get processed in the sub-style files.

theme.sty	colortheme.sty	color
lang	fonttheme.sty	
cjk		
gbt	outertheme.sty	pattern,navigation,lang
other settings	innertheme.sty	pattern,color,lang

3.2 Color

The color style file `beamercolorthemeSJTUBeamermin.sty` is the color setup of the template. Most color schemes are derived from the basic color of SJTU VI[10]. And to adapt the color definitions of beamer, the corresponding interface is mapped, see 17.2 in [3].

```
interface            color=         red      blue
palette primary      cprimary       #004098  #9E1F36
palette secondary    csecondary     #298626  #F28101
palette tertiary     ctertiary      #004D4B  #FED201
palette quanternary  cquanternary   #FFFFFF  #000000
```

As it is mapped to those beamer interfaces, to use the color, you have to declare the color struct first by

```
\usebeamercolor{palette primary}
\color{palette primary.bg}
```

or simply

```
\usebeamercolor[bg]{palette primary}
```

However, there are scenarios where you cannot put temporary variables in some package options since it expands to `\color{\color{mycolor}}`. In this complex case, the redefinition of those standard colors is required. And that's the reason why `innertheme.sty` gets `color`.

3.3 Font

The font style file `beamerfontthemeSJTUBeamermin.sty` provides the font style of the beamer. In SJTUBeamer MIN, serif math font is used by

```
\usefonttheme{professionalfonts}
```

which will tell beamer not to meddle with the specific font (in this case, math font) to the sans serif one.

It is especially useful if you don't want to create more compilation errors since some engine doesn't support sans serif math font. The workaround for that is to introduce the package below:

```
\RequirePackage[eulergreek]{sansmath}
```

And SJTUBeamer MIN does both.

3.4 Outer

The outer style file `beamerouterthemeSJTUBeamermin.sty` contains the layout of frames. The recommended setup is as follows:

Components	SJTUBeamer MIN
head- and footline	●
sidebars	
logo	●
frame title	●

3.5 Inner

The inner style file `beamerinnerthemeSJTUBeamermin.sty` will customize the main components.

Components	SJTUBeamer MIN
Title and part pages	●
Itemize	●
Enumerate	
Description	
Block	
Theorem and proof	
Figures and tables	●
Footnotes	●
Bibliography entries	

Outer theme and inner theme are the core files for SJTUBeamer MIN , which will be discussed in the following content.

4 Compatibility

Since the vision of LaTeX is to build an open-source typesetting system for multi-platforms and beamer is on top of that to create an easy-to-configure interface on building presentations, SJTUBeamer MIN follows the footstep to make its best on compatibility.

4.1 Beamer Interface

Beamer has designed a system of modern interfaces for those theme creators. SJTUBeamer MIN has already followed the modular architecture, as is shown in Section 3.

And there are more APIs in beamer for each corresponding theme style. There are mainly three ways to modify a theme:

1. **Want to use presets.** Read Part III in the documentation of `beamer` package [3]. You can acquire the doc by the terminal command:

   ```
   texdoc beamer
   ```

   Then, you could choose to use some preset theme, or call the macro to control the appearance of each component.

2. **Want a complete modification.** Read the source code of `beamer` package [3]. If no additional theme is used, `beamer` will assume you are creating a theme from `default`. And refer to the corresponding theme file suffixed by `default` will give you the bottom mechanism to implement components.

3. **Want to solve difficult problems.** Go to TEX Stack Exchange [11] for help. Always search before you ask. Then you could probably find some patches and magical formulas to tackle the issue since TEX is a Turing-complete language.

4.2   Mainstream Packages

Mainstream LaTeX packages are used to make sure the choice on marcos is maintained currently. Since some engine doesn't support GhostScript well (*e.g.* XeLaTeX), SJTUBeamer MIN (as well as beamer) uses PGF as the backend for graphics in PostScript. And half of the jobs are done on graphics to implement the requirements of VI.

SJTUBeamer MIN doesn't use too many rasterized pictures, since they are not flexible. You could get the Adobe Illustrator files on VI website[10]. SJTU VI goes minimalism so that it could be implemented by package TikZ (which is on top of PGF). You could almost draw any vectorized shapes by referring to TikZ documentation [12]. In short, TikZ uses node-edge system to create graphs and many Computer Science pictures can be drawn in such a system[13]. And if you don't want to mess around with the thousand pages of documentation, TikZ could help you create that in a WYSIWYG(what you see is what you get) way[14], which is a tool to make drafts on patterns.
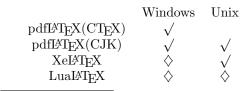
SJTUBeamer MIN also uses additional packages like PGFPLOTS and PGF-PLOTSTABLE to draw highly personalized statistic graphs and layout table from CSV (Comma-Seperated Values) respectively. As is mentioned, the author created a tool PGFPLOTSEDT to help such graphs in a interactive way[9].

Code blocks are drawn by package tcolorbox, which is also a powerful toolkit to make customized boxes[15]. This is almost the most elegant way to make colorful boxes in the current LaTeX system.

Some of the packages have been studied by author in LaTeX Sparkle Project[4]. You can find some lead-in materials to get familiar.

4.3 Engine Support

To be clear, SJTUBeamer MIN is not adapt to all kinds of compliers in the current LaTeX world.

	Windows	Unix
pdfLaTeX(CTeX)	√	
pdfLaTeX(CJK)	√	√
XeLaTeX	◇	√
LuaLaTeX	◇	◇

*√ is fully available, while ◇ will have font issues.

SJTUBeamer MIN make its effort on engine support in the following ways:

1. **Use beamer interface.** As is mentioned in Section 4.1, SJTUBeamer MIN will not create its macro unless there is no substitute in the current version of beamer or it is a common method to implement some features. A good example for this is to make a bottom page, SJTUBeamer MIN mimicked `\maketitle` command to implement `\makebottom` command. A good outcome is that the style file could be separately used with low coupling.

2. **Use mainstream packages.** Mentioned in Section 4.2, mainstream packages are widely accepted in many engines. Some top-level marcos are used to increase the readability of the source code, i.e., PGF is lengthy and hard to be maintained.

3. **Use old-fashioned TeX code.** If there is a nice way to implement in TeX, then go TeX. TeX is a box-based typesetting system, which may be mentioned in many Computer Science books. And LaTeX is on top of that to provide clear-to-read macros. In some scenarios, the native `\vbox` and `\hbox` command could help calculate the position of characters in a more controllable way. But it is certainly painful to learn. The TeX Book[16] is the classic to learn that, but Notes On Programming in TeX[17] is more recommended in modern LaTeX.

5 Principles

Now, you may still be confused about how to create a beamer template. Here is a good material about it for a lead-in[18], which provides a brief overview. And this part is only focusing on the implementation of SJTUBeamer MIN.

5.1 Entry

Create an option by `\DeclareOptionBeamer{}{}`, then set the defualt setting before `\ProcessOptionBeamer` by `\ExcuteOptionsBeamer{}`. Only add neces-

sary options for wider use, since users don't like reading documentations.

<div align="right">theme</div>

```
43  \DeclareOptionBeamer{gbt}{\def\beamer@SJTUBeamermin@gbt{#1}}
44  \def\beamer@SJTUBeamermin@gbttrue{true}%
45  \def\beamer@SJTUBeamermin@gbtfalse{false}%
46  \def\beamer@SJTUBeamermin@gbtbibtex{bibtex}%
47
48  \ExecuteOptionsBeamer{
49      navigation=tools,
50      cjk=false,
51      lang=cn,
52      color=blue,
53      pattern=title,
54      gbt=false,
55  }
56  \ProcessOptionsBeamer
```

Disables a warning by redefining a package hyperref macro [19], which package silence cannot handle.

<div align="right">theme</div>

```
59  \def\Hy@WarnOptionDisabled#1{
60      \def\next{#1}%
61      \ifx\next pdfauthor %
62          \ifx\next driverfallback %
63          \else
64          \Hy@Warning{%
65              Option '#1' has already been used,\MessageBreak
66              setting the option has no effect%
67          }
68      \fi%
69  }
```

Redirect \cite command when using biblatex.

<div align="right">theme</div>

```
93      \def\cite#1{
94          \footfullcite{#1}
```

Specify presentation mode. Enable compress option on beamer to avoid multiline navigation dots and process the sub-styles in order.

<div align="right">theme</div>

```
101  \mode<presentation>
102  \beamer@compresstrue
103  \usecolortheme{SJTUBeamermin}
104  \usefonttheme{SJTUBeamermin}
105  \useoutertheme{SJTUBeamermin}
106  \useinnertheme{SJTUBeamermin}
```

5.2 Pattern

Mono-unit: stamp. The newest version of Ti*k*Z provides the interface to user-define a pattern[12]. Obeying Compatibility philosophy, use `\pgfkeyvalueof` interface to get parameters in a standard way. The unit is first tested in a standalone file and previewed by Ti*k*ZEdt.

innertheme

```
52  \tikzdeclarepattern{
53    name=stamp,
54    parameters={
55      \pgfkeysvalueof{/pgf/pattern keys/size},
56      \pgfkeysvalueof{/pgf/pattern keys/xshift},
57      \pgfkeysvalueof{/pgf/pattern keys/yshift},
58    },
59    defaults={
60      size/.initial = 5pt,
61      xshift/.initial = 0pt,
62      yshift/.initial = 0pt,
63    },
64    bottom left={(
65      -0.5*\pgfkeysvalueof{/pgf/pattern keys/size}+\pgfkeysvalueof{/pgf/
             pattern keys/xshift},
66      -0.4*\pgfkeysvalueof{/pgf/pattern keys/size}+\pgfkeysvalueof{/pgf/
             pattern keys/yshift}
67    )},
68    top right={(
69      0.5*\pgfkeysvalueof{/pgf/pattern keys/size}+\pgfkeysvalueof{/pgf/
             pattern keys/xshift},
70      0.4*\pgfkeysvalueof{/pgf/pattern keys/size}+\pgfkeysvalueof{/pgf/
             pattern keys/yshift}
71    )},
72    tile size={(
73      \pgfkeysvalueof{/pgf/pattern keys/size},
74      0.8*\pgfkeysvalueof{/pgf/pattern keys/size}
75    )},
76    code={
77      \def\s{\pgfkeysvalueof{/pgf/pattern keys/size}}%
78      \tikzset{x=0.5*\s,y=0.2*\s}
79      \fill[xshift=\pgfkeysvalueof{/pgf/pattern keys/xshift},
80        yshift=\pgfkeysvalueof{/pgf/pattern keys/yshift}]
81        (-0.25*\s,0)
82        -- (-0.17*\s,0.06*\s)
83        -- (-0.17*\s,0.1*\s)
84        -- (0.17*\s,0.1*\s)
85        -- (0.17*\s,0.06*\s)
86        -- (0.25*\s,0)
87        -- (0.17*\s,-0.06*\s)
88        -- (0.17*\s,-0.1*\s)
89        -- (-0.17*\s,-0.1*\s)
```

```
90        -- (-0.17*\s,-0.06*\s) -- cycle;
91    }
92  }
```

Array: stamparray.   Provide a marco to create an array of stamp mono-unit. Notice TeX is not good at handling parameters. Always remember to store it into a temporary variable. Register `\pgfmathresult` will store the result of `\pgfmathparse`.

```
103  \providecommand{\stamparray}[3]{
104    % #1: pattern size
105    % #2: starting point
106    % #3: ending point
107      \usebeamercolor{palette primary}
108      \fill [pattern={stamp[size=#1]},pattern color=bg!50!fg] #2 rectangle
            #3;
109      \def\s{#1}%
110      \pgfmathparse{0.5*\s}\let\xs=\pgfmathresult%
111      \pgfmathparse{-0.4*\s}\let\ys=\pgfmathresult%
112      \fill [pattern={stamp[size=#1,xshift=\xs, yshift=\ys]},pattern color=
            bg!50!fg] #2 rectangle #3;
113  }
```

Line: stampline.   Declare a decoration to make a loop stampline. Notice that `auto corner on length` is open to avoid spikes where the state hasn't meet `final` yet.

```
331  \pgfdeclaredecoration{stampline}{initial}
332  {
333    \state{initial}[
334      width=\pgfdecorationsegmentlength,
335      auto corner on length=\pgfdecorationsegmentlength]
336    {
337      \def\l{\pgfdecorationsegmentlength}%
338      \pgfpathlineto{\pgfpoint{0.25*\l}{0pt}}
339      \pgfpathlineto{\pgfpoint{0.33*\l}{0.06*\l}}
340      \pgfpathlineto{\pgfpoint{0.33*\l}{0.1*\l}}
341      \pgfpathlineto{\pgfpoint{0.67*\l}{0.1*\l}}
342      \pgfpathlineto{\pgfpoint{0.67*\l}{0.06*\l}}
343      \pgfpathlineto{\pgfpoint{0.75*\l}{0pt}}
344      \pgfpathlineto{\pgfpoint{\l}{0pt}}
345    }
346    \state{final}
347    {
348      \pgfpathlineto{\pgfpointdecoratedpathlast}
349    }
350  }
```

12

Fade and tile. Use a composition fade of right and center to make a mask on the array, avoiding the collision on the text.

<div align="right">innertheme</div>

```
93   \tikzfading[
94     name=center fade,
95     inner color=transparent!0,
96     outer color=transparent!15
97   ]
98   \tikzfading[
99     name=fade right,
100    left color=transparent!0,
101    right color=transparent!100
102  ]
```

`\the\paperwidth` will give you the width of the paper, the same goes `\the\paperheight`. A PGF length could be calculated by PGFMATH[12].

<div align="right">innertheme</div>

```
125      \begin{tikzpicture}[overlay]
126        \stamparray{20pt}{(-0.2*\the\paperwidth,-1*\the\paperheight)}{(1*\
               the\paperwidth, 0.2*\the\paperheight)}
127        \fill [bg,path fading=fade right] (-0.2*\the\paperwidth,-1*\the\
               paperheight) rectangle (1*\the\paperwidth, 0.2*\the\paperheight
               );
128        \fill [bg,path fading=center fade,xshift=-10pt,yshift=-20pt] (0.2*\
               the\paperwidth,0) circle [radius=\the\paperwidth];
129      \end{tikzpicture}
```

Bottom Page. TikZ provides `\clip` to make clipping and `scope` environment to make a local zone. It is handly on boolean calculation with the combination property.

<div align="right">innertheme</div>

```
443      \usebeamercolor{palette primary}
444      \usebeamercolor{palette secondary}
445      \begin{tikzpicture}[overlay,yshift=-80pt]
446        \def\w{\the\paperwidth}%
447        \def\h{\the\paperheight}%
448        \fill [palette primary.bg] (-0.2*\w,-1*\h) rectangle (1*\w, 0.5*\h);
449        \begin{scope}[fill=palette primary.bg!50!black,fill opacity=0.15]
450          \clip (0.63*\w,1.05*\h) circle (1*\h);
451          \fill (0.14*\w,-0.95*\h) circle (1.67*\h);
452        \end{scope}
453        \begin{scope}[fill=palette secondary.bg!50!palette primary.bg!70!
               white,fill opacity=0.15]
454          \clip[xshift=26pt] (0.95*\w,-0.67*\h) circle (1.17*\h);
455          \fill
456            (0.14*\w,-0.95*\h) circle (1.67*\h)
457            (0.63*\w,1.05*\h) circle (1*\h);
```

<div align="center">13</div>

```
458        \end{scope}
459      \end{tikzpicture}
```

5.3 Gadgets

Logo and institute. `resizebox` is used to adapt to all size of logo into 1cm height one[20]. And it is the same in outer theme to make a 0.7cm logo. The institute is in TEX code for typesetting. `\beamer@shortinstiture` meta is used to avoid compressing on `\par`. Super small font is indicated by `fontsize`.

<div align="right">innertheme</div>

```
136      \begin{beamercolorbox}{logo}
137        \vskip8pt
138        \hbox{
139          \hskip4.5pt{\resizebox{!}{1cm}{\insertlogo}}
140          \ifx\insertinstitute\@empty%
141          \else
142            {\hskip3pt \vrule width0.5pt}\hskip7pt
143            \ifx\beamer@SJTUBeamermin@lang\beamer@SJTUBeamermin@langcn%
144              \vbox{
145                \fontsize{13pt}{0pt}\selectfont
146                \insertinstitute
147                \par\noindent\vskip0.15em
148                \fontsize{5pt}{0pt}\selectfont
149                \textsc{\insertshortinstitute}
150                \baselineskip 3.2pt
151                \par~
152              }
153            \else%
154              \vbox to 1cm{
155                \vfill
156                \vbox{
157                  \offinterlineskip
158                  \noindent \strut
159                  \baselineskip 0pt \lineskip -2pt
160                  \scriptsize\textsc{\beamer@shortinstitute}
161                  \strut
162                }
163                \vfill
164              }
165            \fi%
166          \fi%
167        }
168        \vskip8pt
169      \end{beamercolorbox}
```

Codeblock. Codeblock is created in tcolorbox. The color is redefined in the inner theme to avoid the bad displaying result.

```
291  \tcbuselibrary{skins}
292  \tcbuselibrary{listingsutf8}
293  \lstset{
294    basicstyle=\ttfamily\small,
295    keywordstyle=\color{cprimary},%
296    stringstyle=\color{csecondary},%
297    commentstyle=\color{ctertiary!80!gray},%
298    columns=flexible,
299    extendedchars=false,
300    showstringspaces=false,
301    showspaces=false,
302    breaklines
303  }
304  \newtcblisting{codeblock}[2]{listing only,
305  listing engine=listings,
306  listing options={
307    language=#1,%
308    numbers=left,
309    numberstyle=\color{cprimary!80}\ttfamily\scriptsize,
310    numbersep=5pt,
311  },
312  enhanced,
313  sharp corners,
314  top=0mm,
315  bottom=0mm,
316  title=#2,
317  colback=cprimary!5,
318  colframe=cprimary!80,
319  overlay={
320      \begin{tcbclipinterior}\fill[cprimary!20]%
321          (frame.south west) rectangle ([xshift=5mm]frame.north west);
322      \end{tcbclipinterior}
323  }
324  }
```

Navigation. Clear the original definition first. Then append the page info to the footline, which could avoid collision on footnote. Hide the navigation info automatically when detecting that it is a part page, since there is a navigation bar in that page.

```
50  \setbeamertemplate{sidebar right}{}
51  \ifx\beamer@SJTUBeamermin@navigation\
        beamer@SJTUBeamermin@navigationsubsections%
52    \addtobeamertemplate{footline}{
53      \vskip 4pt
54      \vbox{}
55      \ifnum\beamer@partstartpage=\c@page %
```

```
56     \else
57       \par\hfill\insertnavigation{0.5\paperwidth}\hspace*{0.1cm}
58     \fi
59     \par
60     \vskip 10pt
61     \vbox{}
62   }{}
63 \else
64   \addtobeamertemplate{navigation symbols}{}{%
65   \usebeamerfont{footline}%
66   \usebeamercolor[fg]{footline}%
67   \hspace{1em}%
68   \raisebox{1.2pt}[0pt][0pt]{\insertframenumber/\inserttotalframenumber\
           hspace*{0.5em}}%
69   }%
70   \addtobeamertemplate{footline}{
71     \ifnum\beamer@partstartpage=\c@page %
72     \else%
73       \hfill%
74       \ifx\beamer@SJTUBeamermin@navigation\
               beamer@SJTUBeamermin@navigationtools%
75         \usebeamertemplate***{navigation symbols}%
76       \else
77         \raisebox{1.2pt}[0pt][0pt]{\usebeamerfont{footline}%
78       \insertframenumber/\inserttotalframenumber\hspace*{0.5em}}%
79       \fi
80     \fi%
81     \hspace*{0.1cm}\par
82     \vskip 4pt
83   }{}
84 \fi%
```

Frame title.   Use \ifbeamer@draftmode to detect draft mode. And the cases for frame title only and with subtitle should have different calculations on the height.

```
112       \usebeamerfont{frametitle}%
113       \ifbeamer@draftmode%
114       \else%
115       \ifx\beamer@SJTUBeamermin@pattern\beamer@SJTUBeamermin@patternall
116         \ifnum\beamer@sectionstartpage=\c@page %
117           \begin{tikzpicture}[overlay]
118             \ifx\insertframesubtitle\@empty%
119               \def\h{-0.11*\the\paperheight}
120             \else
121               \def\h{-0.125*\the\paperheight}
122             \fi
123             \usebeamercolor{palette primary}
```

```
124          \stamparray{20pt}{(-0.05*\the\paperwidth,\h)}{(\the\paperwidth
                ,0.05*\the\paperheight)}
125          \fill [bg,path fading=fade left little] (-0.05*\the\paperwidth
                ,\h) rectangle (\the\paperwidth,0.05*\the\paperheight);
126        \end{tikzpicture}
127      \fi
128    \fi
129  \fi%
```

And affect the final closing.

<div align="right">outertheme</div>

```
145    \ifx\insertframesubtitle\@empty\vskip-2.5ex%
146    \else\vskip-3.5ex\fi%
```

References

[1] "MiKTEX." [Online]. Available: https://miktex.org/

[2] "Perl." [Online]. Available: https://www.perl.org/

[3] T. Tantau, J. Wright, and V. Miletić, *The beamer class: User Guide for version 3.59.*, Jul. 2020. [Online]. Available: https://github.com/josephwright/beamer

[4] Log Creative, "LATEX Sparkle Project." [Online]. Available: https://logcreative.github.io/LaTeXSparkle/

[5] TUG, "Upgrade from TEX Live 2020 to 2021." [Online]. Available: http://www.tug.org/texlive/upgrade.html

[6] Log Creative. SJTUBeamermin. [Online]. Available: https://www.overleaf.com/latex/templates/sjtubeamermin/shxnnnjjgqvp

[7] xu-cheng, "LATEX-action." [Online]. Available: https://github.com/xu-cheng/latex-action

[8] Log Creative, "AutoBeamer." [Online]. Available: https://github.com/LogCreative/AutoBeamer

[9] ——, "PGFPlotsEdt." [Online]. Available: https://github.com/LogCreative/PGFPlotsEdt/

[10] Shanghai Jiao Tong University, *SJTU VI Manual*, Apr. 2016. [Online]. Available: https://vi.sjtu.edu.cn/

[11] "TeX Stack Exchange." [Online]. Available: https://tex.stackexchange.com/

[12] T. Tantau, *The TikZ and* PGF *Packages*, Institut für Theoretische Informatik, Universität zu Lübeck, Oct. 2020. [Online]. Available: https://github.com/pgf-tikz/pgf

[13] Paul, "Tunight: A Drawing Guide for TikZ," Oct. 2020. [Online]. Available: https://tuna.moe/event/2020/tikz/

[14] Thomas and Julian, "TikzEdt – A semigraphical TikZ-editor," Mar. 2014. [Online]. Available: http://www.tikzedt.org/

[15] T. F. Sturm, *tcolorbox: manual for version 4.42*, Institut für Mathematik und Informatik, Universität der Bundeswehr München, D-85577 Neubiberg, Oct. 2020. [Online]. Available: https://github.com/T-F-S/tcolorbox

[16] Knuth and D. Ervin, *The TEXbook*. American Mathematical Society and Addison-Wesley Publishing Company, May 1991. [Online]. Available: https://github.com/seuliang/the-texbook-cn

[17] Dr. Christian Feuersänger, *Notes On Programming in TEX*, Feb. 2020. [Online]. Available: https://mirrors.sjtug.sjtu.edu.cn/CTAN/graphics/pgf/contrib/pgfplots/doc/TeX-programming-notes.pdf

[18] "Tunight: TeX Workshop 2017," TUNA, Apr. 2017. [Online]. Available: https://www.youtube.com/watch?v=fysg1IaCq2s&t=5328s

[19] S. Rahtz and H. Oberdiek, *Hypertext marks in LATEX*, The LATEX3 Project, May 2020. [Online]. Available: https://github.com/latex3/hyperref

[20] K. Reckdahl, *Using Import graphics in LATEX 2ε*, Dec. 1997. [Online]. Available: https://www.latexstudio.net/archives/10738