

# Developer's Guide on SJTUBeamer MIN

Log Creative

1.0 August 15, 2021

## Contents

<b>1</b>	<b>Preface</b>	<b>3</b>
<b>2</b>	<b>Build</b>	<b>3</b>
<b>3</b>	<b>Compilation</b>	<b>4</b>
3.1	MiKTeX . . . . .	4
3.2	T <sub>E</sub> X Live . . . . .	4
3.3	Boost Up . . . . .	6
<b>4</b>	<b>Modular Architecture</b>	<b>6</b>
4.1	Theme . . . . .	6
4.2	Color . . . . .	7
4.3	Font . . . . .	8
4.4	Outer . . . . .	8
4.5	Inner . . . . .	8
<b>5</b>	<b>Compatibility</b>	<b>9</b>
5.1	Beamer Interface . . . . .	9
5.2	Mainstream Packages . . . . .	10
5.3	Engine Support . . . . .	10
<b>6</b>	<b>Implementation</b>	<b>11</b>
6.1	Color Theme . . . . .	11
6.1.1	Option Declartion . . . . .	11
6.1.2	Beamer Color Interface . . . . .	11
6.2	Font Theme . . . . .	13
6.3	Inner Theme . . . . .	13
6.3.1	Package Dependencies . . . . .	13
6.3.2	Option Declaration . . . . .	13
6.3.3	Shape Dependencies . . . . .	14
6.3.4	Title Page . . . . .	14

6.3.5	Part Page	17
6.3.6	Section Page and Subsection Page	18
6.3.7	Itemize Environments	19
6.3.8	Block Environments	19
6.3.9	Figures	20
6.3.10	Tables	22
6.3.11	Footnotes	22
6.3.12	Bottom Page	22
6.4	Outer Theme	24
6.4.1	Option Declartion	24
6.4.2	Sidebar	25
6.4.3	Shape Dependencies	26
6.4.4	Frame Title	26
6.5	Parent Theme	29
6.5.1	Option Declaration	29
6.5.2	Option Execution	31
6.6	Color Definition	32
6.7	Logo	33
6.7.1	Load TikZ	33
6.7.2	Shape Declarations	33

# 1 Preface

SJTUBeamer `MIN` is a presentation template based on `beamer` package in  $\text{\LaTeX}$ , to fulfill the enthusiasm of those SJTU users to present their content nicely, benefiting from the technology of  $\text{\TeX}$  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

<code>MIN</code>	- <i>minimal</i> :	minimal work set of SJTU VI.
<code>MIN</code>	- <i>minimalism</i> :	designed in the style of minimalism.
<code>MIN</code>	- <i>minimum</i> :	minimum shapes to show your content.

# 2 Build

To make a CTAN package, a modern `l3build` method is adopted for building the package. And all source code has been refactored.

To build the package:

```
l3build ctan
```

To install the package:

```
l3build install
```

Sometimes, you have to indicate the install directory as follows:

```
l3build install --texmfhome path/to/install
```

To bump version

```
l3build tag 1.0
```

Since `l3build` is still under development, the instability may be introduced to some operating systems like Windows. In this case, use

```
cd source
latex beamerthemesjtubeamermin.ins
```

and move the output file to your installation directory or the directory contains your working file.

It is soon to have a CTAN distribution. At that time, if you are using TeX Live:

```
tlmgr install sjtubeamermin
```

Or use MiKTeX:

```
\usepackage{sjtubeamermin} % trigger the installation
```

and you are ready to go!

### 3 Compilation

Most problems come from L<sup>A</sup>T<sub>E</sub>X compilation. The required packages are in the following list.

pgfplots	tikz	xcolor
pgfplotstable	sansmath	tcolorbox
ctex	biblatex	beamer

The detailed description is documented below.

#### 3.1 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
```

#### 3.2 T<sub>E</sub>X Live

Since some packages are not default installed in the full release of T<sub>E</sub>X 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{}
\end{CJK}
```

```

\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<sup>[5]</sup>:

```
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
```

### 3.3 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.

## 4 Modular Architecture

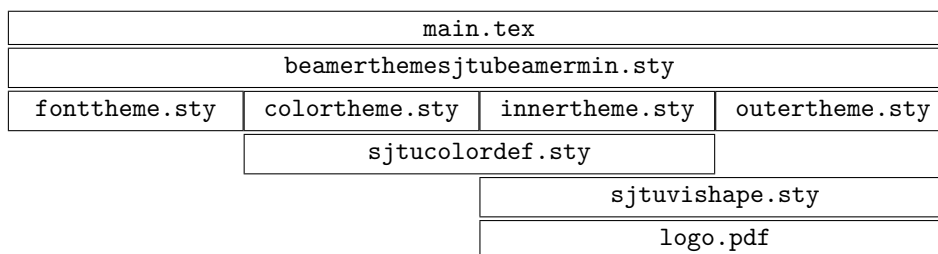
By the recommendation from `beamer` package[\[3\]](#), `SJTUBeamer` `[MIN]` uses the same modular architecture to build the template. Like it is in Java, to let the `beamer` template locate your theme, the style file has to be in the standard name.

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}}`.

### 4.1 Theme

The main theme file `beamerthemeSJTUBeamermin.sty` is the entry point of the theme template. For users, after acquiring the `beamer` package, `\usetheme` com-

.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.
sjtucolordef.sty	Color definition from SJTU VI.
sjtuvishape.sty	VI Shape definition from SJTU VI.



mand 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	outertheme.sty	pattern,navigation,lang
gbt	innertheme.sty	pattern,color,lang
other settings		

And this version meets the standingfree criteria. All source files could be used separately from version 1.0.

## 4.2 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].

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

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

```
\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`.

### 4.3 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.

### 4.4 Outer

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

### 4.5 Inner

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

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



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

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

## 5 Compatibility

Since the vision of L<sup>A</sup>T<sub>E</sub>X 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.

### 5.1 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 4.

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  $\text{\TeX}$  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  $\text{\TeX}$  is a Turing-complete language.

## 5.2 Mainstream Packages

Mainstream  $\text{\LaTeX}$  packages are used to make sure the choice on marcos is maintained currently. Since some engine doesn't support **GhostScript** well (*e.g.*  $\text{\XeLaTeX}$ ), **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, **TikZEdt** 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 **PGFPLOTSTABLE** to draw highly personalized statistic graphs and layout table from CSV (Comma-Separated Values) respectively. As is mentioned, the author created a tool **PGFPLOTSEDIT** to help such graphs in an interactive way[9].

Code blocks are drawn by package **tclobox**, 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  $\text{\LaTeX}$  system.

Some of the packages have been studied by author in  $\text{\LaTeX}$  Sparkle Project[4]. You can check that out to learn more.

## 5.3 Engine Support

To be clear, **SJTUBeamer** [MIN] is not adapt to all kinds of compilers in the current  $\text{\LaTeX}$  world.

	Windows	Unix
$\text{\pdfLaTeX}(\text{\CTeX})$	✓	
$\text{\pdfLaTeX}(\text{\CJK})$	✓	✓
$\text{\XeLaTeX}$	◇	✓
$\text{\LuaLaTeX}$	◇	◇

\*✓ 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 5.1, 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 5.2, mainstream packages are widely accepted in many engines. Some top-level macros are used to increase the readability of the source code, i.e., PGF is lengthy and hard to be maintained.
3. **Use old-fashioned T<sub>E</sub>X code.** If there is a nice way to implement in T<sub>E</sub>X, then go T<sub>E</sub>X. T<sub>E</sub>X is a box-based typesetting system, which may be mentioned in many Computer Science books. And L<sup>A</sup>T<sub>E</sub>X 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 T<sub>E</sub>X Book[16] is the classic to learn that, but Notes On Programming in T<sub>E</sub>X[17] is more recommended in modern L<sup>A</sup>T<sub>E</sub>X.

## 6 Implementation

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`.

### 6.1 Color Theme

#### 6.1.1 Option Declaration

Color theme gets the `color` option only to select different color scheme.

```
1 \DeclareOptionBeamer{color}{\def\beamer@sjtubeamermin@color{#1}}
2 \def\beamer@sjtubeamermin@colorblue{blue}%
3 \def\beamer@sjtubeamermin@colorred{red}%
4 \ExecuteOptionsBeamer{color=blue}
5 \ProcessOptionsBeamer
```

#### 6.1.2 Beamer Color Interface

Load the common color library for sjtubeamermin.

```
6 \RequirePackage{sjtucolordef}
```

Map the defined color in `sjtucolordef` to the interface of beamer color. Especially, the `structure` interface could not derived from the color palette.

```
7 \setbeamercolor{palette primary}{bg=cprimary,fg=white}
8 \setbeamercolor{palette secondary}{bg=csecondary,fg=white}
```

```

9 \setbeamercolor{palette tertiary}{bg=ctertiary,fg=white}
10 \setbeamercolor{palette quanternary}{bg=,fg=cquanternary}
11 \setbeamercolor{structure}{fg=cprimary}

```

This part defines the color scheme of title.

```

12 \setbeamercolor{background canvas}{bg=white}
13 \setbeamercolor{logo}{use={palette primary},bg=,fg=palette primary.fg}
14 \setbeamercolor{normal text}{fg=black,bg=black!40}
15 \setbeamercolor*{block title}{parent=structure}
16 \setbeamercolor{titlelike}{parent={palette primary}}
17 \setbeamercolor{title}{fg=cprimary,bg=}
18 \setbeamercolor{subtitle}{fg=csecondary,bg=}

```

This part defines the color of block title.

```

19 \setbeamercolor{block title}{fg=white,bg=cprimary!90}
20 \setbeamercolor{block title alerted}{use=alerted text,
21 fg=white,bg=csecondary}
22 \setbeamercolor{block title example}{use=example text,
23 fg=cquanternary,bg=ctertiary}

```

This part defines the color of block body.

```

24 \setbeamercolor{block body}{parent=normal text,use=block title,
25 bg=block title.bg!30}
26 \setbeamercolor{block body alerted}{parent=normal text,
27 use=block title alerted,bg=block title alerted.bg!30}
28 \setbeamercolor{block body example}{parent=normal text,
29 use=block title example,bg=block title example.bg!30}

```

This part defines the color of footline.

```

30 \setbeamercolor{section in head/foot}{use={palette primary},
31 fg=palette primary.bg,bg=}

```

This part defines the color of part page, section page, and subsection page.

```

32 \setbeamercolor{part title}{parent={palette primary}}
33 \setbeamercolor{section title}{parent={palette secondary}}
34 \setbeamercolor{subsection title}{parent={palette tertiary}}

```

Set the emphasized color and redefine the emphasizing command to make the text both italic for ASCII character and colored in the middle color of cprimary and csecondary.

The redefinition is required since beamer class has redefined the `\emph` command to make it not nested. According to LearnLaTeX.org, the emphasized color is defined to make contrast in presentation.

For ASCII character, the italic part dominates, as it is quite different from the normal roman font. As for chinese character, the color part dominates, since it is often in bolder shape and changing to other font will make the layout messy.

```

35 \setbeamercolor{emph}{use={palette primary,palette secondary},
36 fg=palette primary.bg!50!palette secondary.bg}
37 \renewcommand<>{\emph}[1]{%
38 {\only#2{\usebeamercolor[fg]{emph}\itshape}#1}%
39 }

```

As is native to beamer, you could also use `\alert` command to highlight the text. The color is redirected to the cprimary.

```
40 \setbeamercolor{alerted text}{use=palette primary,fg=palette primary.bg}
```

## 6.2 Font Theme

Use `professionalfonts` font theme to compress all formula environments, which is in serif font style.

```
41 \usefonttheme{professionalfonts}
```

Use `sansmath` package to support sans serif math font in some blocks, e.g., PGFPlots.

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

Set the font size to normal size for the number indication in part page, section page, and subsection page.

```
43 \setbeamerfont{part name}{size=\normalsize}
```

```
44 \setbeamerfont{section name}{size=\normalsize}
```

```
45 \setbeamerfont{subsection name}{size=\normalsize}
```

Set the font size in the footnote to footnotesize.

```
46 \setbeamerfont{footnote}{size=\footnotesize}
```

## 6.3 Inner Theme

A beamer inner theme dictates the style of the frame elements traditionally set in the “body” of each slide. These include:

- title, part, and section pages;
- itemize, enumerate, and description environments;
- block environments including theorems and proofs;
- figures and tables; and
- footnotes and plain text.

### 6.3.1 Package Dependencies

```
47 \RequirePackage{pgfplots}
```

```
48 \RequirePackage{array}
```

```
49 \RequirePackage{colortbl}
```

```
50 \RequirePackage{booktabs}
```

```
51 \RequirePackage{pgfplotstable}
```

```
52 \RequirePackage{tcolorbox}
```

```
53 \RequirePackage{multicol}
```

### 6.3.2 Option Declaration

`lang` Specify the language of this beamer, which will affect the version of the loaded logo.

```

54 \DeclareOptionBeamer{lang}{\def\beamer@sjtubeamermin@lang{#1}}
55 \def\beamer@sjtubeamermin@langcn{cn}%
56 \def\beamer@sjtubeamermin@langen{en}%

```

**pattern** The pattern mode, which will affect the pattern generation in the title page.

```

57 \DeclareOptionBeamer{pattern}{\def\beamer@sjtubeamermin@pattern{#1}}
58 \def\beamer@sjtubeamermin@patternnone{none}%
59 \def\beamer@sjtubeamermin@patternntitle{title}%
60 \def\beamer@sjtubeamermin@patternall{all}%

```

**color** The selected color theme, which will affect the color in the title page, bottom page and the inner highlighter.

```

61 \DeclareOptionBeamer{color}{\def\beamer@sjtubeamermin@color{#1}}
62 \def\beamer@sjtubeamermin@colorblue{blue}%
63 \def\beamer@sjtubeamermin@colorred{red}%

```

The default default setting will get executed here before the settings defined by the user got processed.

```

64 \ExecuteOptionsBeamer{
65   lang=cn,
66   color=blue,
67   pattern=all
68 }
69 \ProcessOptionsBeamer

```

### 6.3.3 Shape Dependencies

Load the shape package from `sjtuvishape`. To provide the logo, stamp array, and stampline (stampbox is not included).

```

70 \RequirePackage{sjtuvishape}

```

### 6.3.4 Title Page

Declare two fadings: center fade and fade right. The center fade provides a radial fading on the right side of the title page. The fade right provides a linear fading to avoid the collision on the text in the left.

```

71 \tikzfading[
72   name=center fade,
73   inner color=transparent!0,
74   outer color=transparent!15
75 ]
76 \tikzfading[
77   name=fade right,
78   left color=transparent!0,
79   right color=transparent!100
80 ]

```

Define the title page template.

```
81 \defbeamertemplate*{title page}{sjtubeamermin}[1] []
82 {
83   \vbox{}
```

The background of the title page is implemented by a TikZ rectangle, which avoids the changing on `background canvas` beamer color.

In this definition environment, you could not change the beamer color. The older version redefines `\maketitle` command and switches the `background canvas` color, which is harmful for decoupling.

Use TikZ rectangle also avoids the unexpected shift because the risk of redefining the internal command is avoided. If there is any text before the title page, the `\maketitle` will start from a new page.

```
84   \usebeamercolor{palette primary}
85   \begin{tikzpicture}[overlay]
86     \fill [palette primary.bg] (-0.2*\the\paperwidth,-1*\the\paperheight)
87       rectangle (1*\the\paperwidth, 0.2*\the\paperheight);
88   \end{tikzpicture}
```

If it is in draftmode, no pattern will get rendered.

```
89   \ifbeamer@draftmode%
```

Otherwise, the fade tile of stamp array will get covered on top of the background rectangle. `stamp array` is defined in `SJTUvishape`. Then, a fade right covers this array layer and a center fade covers the previous result.

```
90   \else%
91     \ifx\beamer@sjtubeamermin@pattern\beamer@sjtubeamermin@patternnone%
92     \else%
93     \begin{tikzpicture}[overlay]
94       \stamparray{20pt}
95       {(-0.2*\the\paperwidth,-1*\the\paperheight)}
96       {(1*\the\paperwidth, 0.2*\the\paperheight)}
97       \fill [bg,path fading=fade right]
98         (-0.2*\the\paperwidth,-1*\the\paperheight) rectangle
99         (1*\the\paperwidth, 0.2*\the\paperheight);
100      \fill [bg,path fading=center fade,xshift=-10pt,yshift=-20pt]
101        (0.2*\the\paperwidth,0) circle [radius=\the\paperwidth];
102    \end{tikzpicture}
103    \fi%
104  \fi%
```

Set a constraint in the vertical mode to make the following contents centered in the middle of the slide.

```
105   \vfill
106   \begin{group}
107     \centering
```

`resizebox` is used to adapt to all size of logo into 1cm height one. And it is the same in outer theme to make a 0.7cm logo. The institute is in  $\text{\TeX}$  code for typesetting. `\beamer@shortinstitute` meta is used to avoid compressing on `\par`, while `\insertinstitute` will force the input to spread on one single line.

The mode to use is depended on the `language` option. Super small font could be made by `fontsize`.

```

108     \usebeamercolor{titlelike}
109     \begin{beamercolorbox}{logo}
110         \vskip8pt
111         \hbox{
112             \hskip4.5pt{\resizebox{!}{1cm}{\insertlogo}}
113             \ifx\insertinstitute\@empty%
114             \else
115                 \ifx\insertlogo\@empty%
116                 \else
117                     {\hskip3pt \vrule width0.5pt}\hskip7pt
118                 \fi
119                 \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
120                     \vbox{
121                         \fontsize{13pt}{0pt}\selectfont
122                         \insertinstitute
123                         \par\noindent\vskip0.15em
124                         \fontsize{5pt}{0pt}\selectfont
125                         \textsc{\insertshortinstitute}
126                         \baselineskip 3.2pt
127                         \par~
128                     }
129                 \else%
130                     \vbox to 1cm{
131                         \vfill
132                         \vbox{
133                             \offinterlineskip
134                             \noindent \strut
135                             \baselineskip 0pt \lineskip -2pt
136                             \scriptsize\textsc{\beamer@shortinstitute}
137                             \strut
138                         }
139                         \vfill
140                     }
141                 \fi%
142             \fi%
143         }
144         \vskip8pt
145     \end{beamercolorbox}

```

Insert title, subtitle, author, and date.

```

146     \begin{beamercolorbox}[sep=8pt,#1]{title}
147         \usebeamercolor[fg]{palette primary}
148         \usebeamerfont{title}\inserttitle\par%
149         \ifx\insertsubtitle\@empty%
150         \else%
151             \vskip0.25em%
152             {\usebeamerfont{subtitle}\insertsubtitle\par}%
153         \fi%

```



```

154 \end{beamercolorbox}%
155 \vskip1em\par
156 \begin{beamercolorbox}[sep=8pt,#1]{author}
157   \usebeamerfont{author}\insertauthor
158 \end{beamercolorbox}
159 \begin{beamercolorbox}[sep=8pt,#1]{date}
160   \usebeamerfont{date}\insertdate
161 \end{beamercolorbox}

```

Here insert the titlegraphic. The node position is set to `above left` to make sure the bottom of the picture is aligned to the bottom of the date line.

```

162 \begin{tikzpicture}[overlay,yshift=0.77em]
163   \node [above left] at (0.88*\the\paperwidth,0)
164     {\usebeamercolor{fg}{titlegraphic}\inserttitlegraphic};
165 \end{tikzpicture}
166 \endgroup
167 \vskip0.5em
168 \vfill
169 }

```

### 6.3.5 Part Page

Define the `part page` beamer template.

```

170 \defbeamertemplate*{part page}{sjtubeamermin}[1] []
171 {
172   \vfill
173   \vskip 8pt
174   \beginingroup

```

Print the number of this part. If it is in Chinese, the translated version is printed.

```

175   \begin{beamercolorbox}[sep=16pt,right,#1]{part title}
176     \hfill\usebeamerfont{part name}
177     \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
178       ~\insertromanpartnumber~
179     \else%
180       \partname~\insertromanpartnumber
181     \fi%
182   \par\vskip4pt
183   \usebeamerfont{part title}\insertpart\par

```

Since navigation bar is packaged, to modify the color, you have to change the `section in head/foot` beamer color. Here, the first move is to save the current color to a temporary variable. After the insertion, the previous color should be restored.

```

184   \hbox to \textwidth{
185     \usebeamerfont{footline}%
186     \setbeamercolor{temp}{fg=section in head/foot.fg}
187     \setbeamercolor{section in head/foot}{use=palette primary,
188       fg=palette primary.fg,bg=}
189     \hfill

```

```

190      \insertnavigation{0.4\textwidth}
191      \hspace*{1cm}
192      \setbeamercolor{section in head/foot}{fg=temp.fg}
193    }
194    \end{beamercolorbox}
195  \endgroup
196  \vfill
197 }

```

Redirect the part command to make a part page.

```

198 \AtBeginPart{
199   \begin{frame}
200     \partpage
201   \end{frame}
202 }

```

### 6.3.6 Section Page and Subsection Page

Define the common `\sectionblock` command to make the section block.

```

203 \def\sectionblock#1{
204   \begin{beamercolorbox}[sep=12pt,right,#1]{section title}
205     \usebeamerfont{section name}
206     \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
207       ~\insertsectionnumber~
208     \else%
209       \sectionname~\insertsectionnumber
210     \fi%
211     \par\vskip4pt
212     \usebeamerfont{section title}\insertsection\par
213   \end{beamercolorbox}
214 }

```

Define the section page beamer template.

```

215 \defbeamertemplate*{section page}{sjtubeamermin}[1] []
216 {
217   \vfill
218   \begin{group}
219     \sectionblock{#1}
220   \end{group}
221   \vfill
222 }

```

Define the subsection page beamer template.

```

223 \defbeamertemplate*{subsection page}{sjtubeamermin}[1] []
224 {
225   \vfill
226   \begin{group}
227     \sectionblock{#1}
228     \begin{beamercolorbox}[sep=8pt,right,#1]{subsection title}
229       \usebeamerfont{subsection name}
230       \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%

```

```

231         ~\insertsubsectionnumber~
232     \else%
233         \subsectionname~\insertsubsectionnumber
234     \fi%
235     \par\vskip 4pt
236     \usebeamerfont{subsection title}\insertsubsection\par
237 \end{beamercolorbox}
238 \endgroup
239 \vfill
240 }

```

### 6.3.7 Itemize Environments

Set the item marker to circle and set the marker for section and subsection in TOC (Table of Contents) to circle.

```

241 \setbeamertemplate{items}[circle]
242 \setbeamertemplate{sections/subsections in toc}[circle]

```

### 6.3.8 Block Environments

Introduce `sjtucolordef` package. The user-defined block environment should use the hard-coded color. Otherwise it will have side effect on displaying.

```

243 \RequirePackage{sjtucolordef}

```

`\highlight` Highlight the given text. Create a primary color background block with white as foreground.

```

244 \newtcbbox{\highlight}[1][cprimary]{
245     on line,
246     arc=0pt,
247     colback=#1,
248     colupper=white,
249     boxrule=0pt,
250     boxsep=0pt,
251     left=4pt,
252     right=4pt,
253     top=2pt,
254     bottom=2pt
255 }

```

`\paragraph` Use `\highlight` macro for making contrast. Since beamer has deleted `\paragraph` macro in this class, this template defines a macro for that to indicate it is another point and more paragraph-like. It is useful for the migration from `article` class.

```

256 \def\paragraph#1{\highlight{#1}~}

```

Introduce the library from `tcolorbox` to make code blocks. `listingsutf8` is used to receive UTF-8 input.

```

257 \tcbuselibrary{skins}
258 \tcbuselibrary{listingsutf8}

```

Declare the basic listing highlighter. `columns` is set to `flexible` to avoid ugly grid alignment. `breaklines` is set to enable line wrapping.

```
259 \lstset{
260   basicstyle=\ttfamily\small,
261   keywordstyle=\color{cprimary},%
262   stringstyle=\color{csecondary},%
263   commentstyle=\color{ctertiary!80!gray},%
264   columns=flexible,
265   extendedchars=false,
266   showstringspaces=false,
267   showspace=false,
268   breaklines
269 }
```

**codeblock** Code block environment is made for presenting code in an obvious way. Two parameters are required. The first parameter is passed to `listing`, which mostly sets the language to highlight, see the `listings` package for more details. And the second parameter receives the title to make.

```
270 \newtcblisting{codeblock}[2][*]{
271   listing only,
272   listing engine=listings,
273   listing options={
274     #1,%
275     numbers=left,
276     numberstyle=\color{cprimary!80}\ttfamily\scriptsize,
277     numbersep=5pt,
278   },
279   enhanced,
280   sharp corners,
281   top=0mm,
282   bottom=0mm,
283   title=#2,
284   colback=cprimary!5,
285   colframe=cprimary!80,
286   overlay={
287     \begin{tcbclipinterior}\fill[cprimary!20]%
288       (frame.south west) rectangle ([xshift=5mm]frame.north west);
289     \end{tcbclipinterior}
290   }
291 }
```

### 6.3.9 Figures

**stampbox** Make a stampbox border, which is a decoration advice from SJTU VI. It has the dependency on `stampline` from `sjtuvishape` package.

```
292 \newtcolorbox{stampbox}[1][cprimary]{%
293   capture=hbox,
294   enhanced,
295   frame empty,
```

```

296 interior empty,
297 sharp corners,
298 top=2pt,bottom=2pt,left=2pt,right=2pt,
299 borderline={4pt}{0pt}{
300   #1,
301   line width=0.5pt,
302   decoration={
303     stampline,
304     segment length=8pt,
305     path has corners=true,
306   },
307   decorate
308 }
309 }

```

Set the default visual theme for PGFPLOTS. The cycle list is set to the current color theme. And lines on the graph is optimized to make it clear for presentation. The predefinition on the height is made to avoid the overfullbox on the vertical side.

```

310 \pgfplotsset{
311   compat=newest,
312   every axis/.style={
313     font=\small\sffamily\sansmath,
314     cycle multi list={
315       mark options={thick}\nextlist
316       mark list\nextlist
317       cprimary,csecondary,ctertiary\nextlist
318     },
319     height=0.5*\the\paperheight,
320     axis line style={
321       cprimary,
322       thin
323     },
324     every tick label/.style={
325       cprimary,
326       font=\small,
327       /pgf/number format/assume math mode=true
328     },
329     grid style={cprimary!10},
330     tick style={cprimary!50},
331     minor tick style={cprimary!30},
332     xlabel style={cprimary},
333     ylabel style={cprimary},
334     zlabel style={cprimary},
335     legend style={
336       draw={cprimary},
337       thin,
338       nodes={cprimary}
339     },
340     thick,

```

```

341   },
342 }

```

### 6.3.10 Tables

Two macros are defined to make the header colored.

```

343 \def\zapcolorreset{\let\reset@color\relax\ignorespaces}
344 \def\colorrows#1{\noalign{\aftergroup\zapcolorreset#1}\ignorespaces}

```

Set the style of PGFPLOTSTABLE. The `\colorrows` macro here is used for making the header colored. The `booktabs` line is used to create a professional look.

```

345 \pgfplotstableset{
346   col sep=comma,
347   every table/.style={
348     font={\small\sffamily},
349   },
350   every head row/.style={
351     before row={
352       \arrayrulecolor{cprimary}
353       \toprule
354       \colorrows{\color{cprimary}}
355     },
356     after row={
357       \midrule
358       \colorrows{\color{black}}
359     },
360   },
361   every last row/.style={
362     after row=\bottomrule
363   },
364 }

```

### 6.3.11 Footnotes

Define the footline beamer template. The format is slightly changed from the original beamer definition.

```

365 \defbeamertemplate*{footnote}{\sjtbeamerman}
366 {
367   \usebeamerfont{footnote}
368   \parindent 0.5em\noindent%
369   \raggedright
370   \hbox to 1.5em{\hfil\insertfootnotemark}\insertfootnotetext\par%
371 }

```

### 6.3.12 Bottom Page

`\bottompage` Define the macro `\bottompage` to create the ending frame.

```

372 \def\bottompage{

```

Enter vertical mode.

```
373 \vbox{}
```

Create the background canvas and the three overlapping circles in the right. Use `scope` to define the influence range. And use `\clip` to make the clipping in the current range.

```
374 \usebeamercolor{palette primary}
375 \usebeamercolor{palette secondary}
376 \begin{tikzpicture}[overlay,yshift=-80pt]
377   \def\w{\the\paperwidth}%
378   \def\h{\the\paperheight}%
379   \fill [palette primary.bg] (-0.2*\w,-1*\h) rectangle (1*\w, 0.5*\h);
380   \begin{scope}[fill=palette primary.bg!50!black,fill opacity=0.15]
381     \clip (0.63*\w,1.05*\h) circle (1*\h);
382     \fill (0.14*\w,-0.95*\h) circle (1.67*\h);
383   \end{scope}
384   \begin{scope}[fill=palette secondary.bg!50!palette primary.bg!70!white,
385     fill opacity=0.15]
386     \clip[xshift=26pt] (0.95*\w,-0.67*\h) circle (1.17*\h);
387     \fill
388       (0.14*\w,-0.95*\h) circle (1.67*\h)
389       (0.63*\w,1.05*\h) circle (1*\h);
390   \end{scope}
391 \end{tikzpicture}
```

Insert the logo in the crossing center of the overlapping circles.

```
392 \vfill
393 \begingroup
394   \raggedleft
395   \resizebox{!}{1cm}{\insertlogo}
```

Inset the “thank you” quote and the title of this beamer. Notice that three `\vfill` divide the frame into three portions with final adjust using `\vskip`.

```
396 \vfill
397 \vskip6em
398 \begin{beamercolorbox}[sep=8pt]{title}
399   \usebeamercolor{fg}{palette primary}
400   \usebeamerfont{title}\noindent
401   \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn
402
403   \else
404     Thank You
405   \fi
406   \vskip1em%
407   \usebeamerfont{subtitle}\insertauthor~$\cdot$~\inserttitle
408 \end{beamercolorbox}%
409 \vfill
410 \vskip3.5em
411 \endgroup
412 }
```

`\makebottom` The standard interface for making the bottom page in this template. Since there is no standard interface in beamer, the macro mimicked `\maketitle` macro to provide such an interface.

```
413 \def\makebottom{
414   \ifbeamer@inframe%i
415   \bottompage%
416   \else%
417   \frame{\bottompage}%
418   \fi%
419 }
```

## 6.4 Outer Theme

A `beamer` outer theme dictates the style of the frame elements traditionally set outside the body of each slide: the head, footline, and frame title.

### 6.4.1 Option Declartion

`lang` Receive the language option.

```
420 \DeclareOptionBeamer{lang}{\def\beamer@sjtubeamermin@lang{#1}}
421 \def\beamer@sjtubeamermin@langcn{cn}%
422 \def\beamer@sjtubeamermin@langen{en}%
```

`pattern` Sets the pattern visibility in the title page and the header of each slide.

```
423 \DeclareOptionBeamer{pattern}{\def\beamer@sjtubeamermin@pattern{#1}}
424 \def\beamer@sjtubeamermin@patternnone{none}%
425 \def\beamer@sjtubeamermin@patterntitle{title}%
426 \def\beamer@sjtubeamermin@patternall{all}%
```

`navigation` Set the style of navigation bar.

`tools` The default navigation tools provided by `beamer` package, with the page number provided.

`subsections` The subsection progress bar, like the headline in `miniframe` outer theme.

`pages` The page number and the total page number only.

```
427 \DeclareOptionBeamer{navigation}{\def\beamer@sjtubeamermin@navigation{#1}}
428 \def\beamer@sjtubeamermin@navigationtools{tools}%
429 \def\beamer@sjtubeamermin@navigationsubsections{subsections}%
430 \def\beamer@sjtubeamermin@navigationpages{pages}%
```

Set up the default options of the outer theme. And then process the setting passed to the outer theme.

```
431 \ExecuteOptionsBeamer{
432   lang=cn,
433   pattern=all,
434   navigation=tools
435 }
436 \ProcessOptionsBeamer
```



## 6.4.2 Sidebar

Clear the original definition of sidebar first. Then append the page info to the footline, which could avoid collision on footnote.

```
437 \setbeamertemplate{sidebar right}{}%
```

If the `navigation` option is set to `subsections`, then by calling `\insertnavigation` method embeded in beamer class, a subsection navigation toolbar could be generated. You could change the width of the subsection navigation bar in the first parameter of `\insertnavigation` command.

Hide the navigation info automatically when detecting that it is a part page, since there is a navigation bar in that page (defined in the inner theme). However, `\ifnum` may introduce some extra spacing, thus the top margin and the bottom margin could be a little bit different.

```
438 \ifx\beamer@sjtubeamermin@navigation\beamer@sjtubeamermin@navigationsubsections%
439 \addtobeamertemplate{footline}{
440   \vskip 4pt
441   \vbox{}
442   \ifnum\beamer@partstartpage=\c@page %
443   \else
444     \par\hfill\insertnavigation{0.4\paperwidth}\hspace*{0.1cm}
445   \fi
446   \par
447   \vskip 10pt
448   \vbox{}
449 }{}
```

Else, the option could be either `tools` or `pages`.

```
450 \else
```

Define the `\pagenumbering` macro to insert both the current page number and the total page number. With the proper font and color setting from `footline` and raise a little bit by a `\raisebox`.

```
451 \def\pagenumbering{
452   \raisebox{1.2pt}[0pt][0pt]{
453     \usebeamerfont{footline}
454     \usebeamercolor{footline}
455     \color{footline.fg!50}
456     \insertframenumber/\inserttotalframenumber
457     \hspace*{0.5em}
458   }
459 }
```

Append the page number info into the navigation symbols, which will be called by the `tools` option.

```
460 \addtobeamertemplate{navigation symbols}{}{%
461   \hspace{1em}%
462   \pagenumbering
463 }%
```

Then, for different option, the visual could be different. As always, the toolbar should be hidden if it is a part page. But for `tools` option, use the `navigation symbols` template defined above. For `pages` option, use the `\pagenumbering` macro only.

```

464 \addtobeamertemplate{footline}{
465   \ifnum\beamer@partstartpage=\c@page %
466   \else%
467     \hfill%
468     \ifx\beamer@sjtubeamermin@navigation\beamer@sjtubeamermin@navigationtools%
469       \usebeamertemplate***{navigation symbols}%
470     \else
471       \pagenumbering%
472     \fi
473   \fi%
474   \hspace*{0.1cm}\par
475   \vskip 4pt
476 }{}
477 \fi%
```

### 6.4.3 Shape Dependencies

Load the shape package from `sjtuvishape`. To provide the logo and stamp array.

```

478 \RequirePackage{sjtuvishape}
```

### 6.4.4 Frame Title

Define the fade left little fading for frame title. To create a mask on the stamp array pattern.

```

479 \tikzfading[
480   name=fade left little,
481   right color=transparent!0,
482   left color=transparent!10
483 ]
```

Define the beamer template `frametitle`.

```

484 \defbeamertemplate*{frametitle}{sjtubeamermin}[1][left]
485 {%
486   \ifbeamercolorempy{bg}{frametitle}{}{\nointerlineskip}%
487   \@tempdima=\textwidth%
488   \advance\@tempdima by\beamer@leftmargin%
489   \advance\@tempdima by\beamer@rightmargin%
490   \begin{beamercolorbox}[sep=0.3cm,#1,wd=\the\@tempdima]{frametitle}
491     \begingroup
492     \usebeamerfont{frametitle}%
493     \ifbeamer@draftmode%
494     \else%
```

If it is not in draft mode, then the pattern on the section start page will get rendered. And the pattern height is the same as that of background color block, depend on whether there is a subtitle on that page.

Notice that it is not defined by the final calculation from LaTeX itself – it is rather hard coded.

TODO: There is a potential risk that if the text is longer than one line, the height could be wrong. That’s the reason why it is only rendered in the section start page – to avoid such edge case as much as possible.

```

495     \ifx\beamer@sjtubeamermin@pattern\beamer@sjtubeamermin@patternall
496     \ifnum\beamer@sectionstartpage=\c@page %
497     \begin{tikzpicture}[overlay]
498     \ifx\insertframesubtitle\@empty%
499     \def\h{-0.11*\the\paperheight}
500     \else
501     \def\h{-0.125*\the\paperheight}
502     \fi
503     \usebeamercolor{palette primary}
504     \stamparray{20pt}
505     {(-0.05*\the\paperwidth,\h)}
506     {(\the\paperwidth,0.05*\the\paperheight)}
507     \fill [bg,path fading=fade left little] (-0.05*\the\paperwidth,\h)
508     rectangle (\the\paperwidth,0.05*\the\paperheight);
509     \end{tikzpicture}
510     \fi
511     \fi

```

Insert title and subtitle and make spacing depend on the existence of subtitle.

```

512     \fi%
513     \vbox{}
514     \ifx\insertframesubtitle\@empty\vskip-2pt%
515     \else\vskip-1ex\fi%
516     \if@tempswa\else\csname beamer@fte#1\endcsname\fi%
517     \strut\insertframetitle\strut\par%
518     {%
519     \ifx\insertframesubtitle\@empty%
520     \else%
521     {
522     \usebeamerfont{framesubtitle}
523     \usebeamercolor[fg]{framesubtitle}
524     \strut\insertframesubtitle\strut\par
525     }%
526     \fi
527     }%
528     \vskip-1ex%
529     \endgroup%

```

Finally, add the logo to the upper right corner. It will be scaled to a 0.7cm height one by using `\resizebox`.

```

530     \raggedleft%
531     \begingroup
532     \ifx\insertframesubtitle\@empty\vskip-2.5ex%
533     \else\vskip-3.5ex\fi%
534     {\resizebox{!}{0.7cm}{\insertlogo}}\hspace*{2ex}%

```

```

535     \endgroup%
536     \ifx\insertframesubtitle\@empty%
537     \else\vskip0.5ex\fi%
538     \if@tempwa\else\vskip-.3cm\fi%
539     \end{beamercolorbox}%
540 }

    A temporary move for max option.
541 \RequirePackage{kvoptions}
542 \RequirePackage{iftex}
543
544 \SetupKeyvalOptions{
545     family=sjtubeamer,
546     prefix=sjtubeamer@
547 }
548 \DeclareBoolOption{bannertitle}
549 \DeclareBoolOption{red}
550 \DeclareBoolOption{blue}
551 \ProcessKeyvalOptions*
552
553 \usetheme{Szeged}
554 \usecolortheme{dolphin}
555
556 \usefonttheme{structurebold}
557 \useinnertheme{circles}
558 \setbeamertemplate{blocks}[rounded]
559
560 \ifsjtubeamer@red
561     \definecolor{SJTUPrimary}{RGB}{200,22,30}
562 \fi
563
564 \ifsjtubeamer@blue
565     \definecolor{SJTUPrimary}{RGB}{0,64,152}
566 \fi
567
568 \setbeamercolor{structure}{fg=SJTUPrimary}
569 \setbeamertemplate{background}{\includegraphics[width=\paperwidth]{vi/sjtu-miao.png}}
570
571 \ifsjtubeamer@red
572     \logo{\includegraphics[height=1.5cm]{vi/sjtu-vi-badge-red.pdf}}
573     \setbeamercolor{banner title page}{fg=SJTUPrimary,bg=white}
574 \fi
575
576 \ifsjtubeamer@blue
577     \logo{\includegraphics[height=1.5cm]{vi/sjtu-vi-badge-blue.pdf}}
578     \setbeamercolor{banner title page}{fg=white,bg=SJTUPrimary}
579 \fi
580
581 \ifsjtubeamer@bannertitle
582     \defbeamertemplate*{title page}{sjtured}[1]{}

```

```

583 {
584   \nointerlineskip%
585   \begin{beamercolorbox}[wd=\paperwidth, ht=\paperheight]{empty}
586     \usebeamercolor{banner title page}%
587     \begin{tikzpicture}
588       \pgfmathsetlengthmacro{\outslant}{\the\paperwidth - 0.8cm}
589       \node[anchor=north west, inner sep=0, outer sep=0] at (0,\the\paperheight){\includegraphics[width=\paperwidth]{}}
590       \fill[color=white] (0,0) rectangle(\the\paperwidth,3.9);
591       \fill[color=bg] (0,0) rectangle(\the\paperwidth,3.88);
592       \node[anchor=north west, text width=.8\paperwidth, fg] at (0.8,3.6){%
593         \ifx\insertsubtitle\@empty%
594           {\huge\bfseries\inserttitle}\\
595         \else%
596           {\Large\bfseries\inserttitle}\\
597           {\small\insertsubtitle}\\
598         \fi%
599       };
600       \node[anchor=south west, text width=.8\paperwidth, fg] at (0.8,0.45){%
601         {\small\insertauthor}\\
602         {\small\insertinstitute}\\
603         {\small\insertdate}%
604       };
605       \node[anchor=south east, inner sep=0, outer sep=0] at (\outslant,0.5){
606         \ifsjtubeamer@red
607           \includegraphics[height=1cm]{vi/sjtu-vi-logo-red.pdf}
608         \fi
609
610         \ifsjtubeamer@blue
611           \includegraphics[height=1cm]{vi/sjtu-vi-logo-white.pdf}
612         \fi
613       };
614     \end{tikzpicture}
615   \end{beamercolorbox}
616 }
617 \fi

```

## 6.5 Parent Theme

The primary job of this package is to load the component sub-packages of the SJTUBeamer `MIN` theme and route the theme options accordingly. It also provides some custom commands and environments for the user.

This declares that the following setup is available for all modes.

```
618 \mode<all>
```

### 6.5.1 Option Declaration

**navigation** Change the appearance of the navigation bar, which will affect in the outer theme.

```

619 \DeclareOptionBeamer{navigation}{
620   \PassOptionsToPackage{navigation=#1}{beamerouterthemesjtubeamermin}

```

621 }

**lang** Set the language of this beamer. Two options are provided:

**cn** Chinese. The loaded logo will be the original one. And the package for chinese character support (CTEXor CJK) will be loaded as well. The bibliography will also get affected.

**en** English. The loaded logo will be the English one.

This option will get passed to both inner and outer theme.

```
622 \DeclareOptionBeamer{lang}{  
623     \def\beamer@sjtubeamermin@lang{#1}  
624     \PassOptionsToPackage{lang=#1}{beamerouterthemesjtubeamermin}  
625     \PassOptionsToPackage{lang=#1}{beamerinnerthemesjtubeamermin}  
626 }  
627 \def\beamer@sjtubeamermin@langcn{cn}%  
628 \def\beamer@sjtubeamermin@langen{en}%
```

**cjk** Choose to use ‘CJK’ package. If this option is open, the document body should be covered by `\begin{CJK}{UTF8}{hei}` and `\end{CJK}`.

```
629 \DeclareOptionBeamer{cjk}{\def\beamer@sjtubeamermin@cjk{#1}}  
630 \def\beamer@sjtubeamermin@cjktrue{true}%  
631 \def\beamer@sjtubeamermin@cjkfalse{false}%
```

**color** Provided two options:

**blue** The default selection.

**red** The recommended theme for non-scientific scenario.

This option will be passed to the color theme and inner theme.

```
632 \DeclareOptionBeamer{color}{  
633     \PassOptionsToPackage{color=#1}{beamercolorthemesjtubeamermin}  
634     \PassOptionsToPackage{color=#1}{beamerinnerthemesjtubeamermin}  
635 }
```

**pattern** Provided three options to affect the pattern in the slides:

**none** No patterns will be generated.

**title** A pattern array will get generated in the title page.

**all** Besides the title page, the frame title of section start page will get a stamp array pattern.

This option will get passed to the outer theme and inner theme.

```
636 \DeclareOptionBeamer{pattern}{  
637     \PassOptionsToPackage{pattern=#1}{beamerouterthemesjtubeamermin}  
638     \PassOptionsToPackage{pattern=#1}{beamerinnerthemesjtubeamermin}  
639 }
```

**gbt** Choose the behaviour of citing.

**false** Use biblatex to cite.

**bibtex** Use bibtex to cite.

**true** Use biblatex-gbt7714-2015 to cite.

```
640 \DeclareOptionBeamer{gbt}{\def\beamer@sjtubeamermin@gbt{#1}}
641 \def\beamer@sjtubeamermin@gbttrue{true}%
642 \def\beamer@sjtubeamermin@gbtfalse{false}%
643 \def\beamer@sjtubeamermin@gbtbibtex{bibtex}%
```

The default default setting will get executed here before the settings defined by the user got processed.

```
644 \ExecuteOptionsBeamer{
645     navigation=tools,
646     cjk=false,
647     lang=cn,
648     color=blue,
649     pattern=title,
650     gbt=false,
651 }
652 \ProcessOptionsBeamer
```

### 6.5.2 Option Execution

Disable the warning from `hyperref` which conflicts the setting in C<sub>T</sub>E<sub>X</sub> or CJK. It has to be manually disabled.

```
653 \RequirePackage{silence}
654 \def\Hy@WarnOptionDisabled#1{
655     \def\next{#1}%
656     \ifx\next pdfauthor %
657         \ifx\next driverfallback %
658             \else
659                 \Hy@Warning{%
660                     Option `#1' has already been used,\MessageBreak
661                     setting the option has no effect%
662                 }\fi%
663     \fi%
664 }
```

Process the option of `lang` and `cjk`. For Chinese typesetting, some translations are needed for CJKutf8 package.

```
665 \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langen%
666 \else
667     \ifx\beamer@sjtubeamermin@cjk\beamer@sjtubeamermin@cjktrue%
668         \RequirePackage{CJKutf8}
669         \renewcommand{\figurename}{ }
670         \renewcommand{\tablename}{ }
```

```

671      \renewcommand{\contentsname}{ }
672      \else%
673          \RequirePackage[UTF8]{ctex}
674      \fi%
675 \fi

```

Process the option of `gbt` to handle the behaviour of citing. If `bibtex` is used, the corresponding bibliography style will get loaded according to `lang` option. Otherwise, set the style of `biblatex` and redirect `\cite` to `\footfullcite`.

```

676 \ifx\beamer@sjtubeamermin@gbt\beamer@sjtubeamermin@gbtbibtex%
677     \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langen%
678         \bibliographystyle{IEEEtran}
679     \else
680         \RequirePackage{gbt7714}
681     \fi
682 \else
683     \ifx\beamer@sjtubeamermin@gbt\beamer@sjtubeamermin@gbttrue%
684         \RequirePackage[style=gb7714-2015]{biblatex}
685     \else
686         \RequirePackage[style=authortitle-comp]{biblatex} %
687     \fi
688     \def\cite#1{
689         \footfullcite{#1}
690     }
691 \fi%

```

To avoid the messiness of Chinese bookmarks.

```

692 \hypersetup{unicode}
693 \RequirePackage{bookmark}
694 \WarningFilter{latexfont}{Font shape}

```

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

```

695 \mode<presentation>
696 \beamer@compresstrue
697 \usecolortheme{sjtubeamermin}
698 \usefonttheme{sjtubeamermin}
699 \useoutertheme{sjtubeamermin}
700 \useinnertheme{sjtubeamermin}

```

The following code is merely an implementation of SJTU VI, which doesn't change the ownership of the design pattern. Any commercial usage should be acknowledged by the related administration of SJTU.

## 6.6 Color Definition

The following color is defined by SJTU VI.

**cprimary** The primary color, which influences the color of title and the background of title page.

**csecondary** The secondary color, which influences the color of subtitle.



**ctertiary** The tertiary color, which provides the color for the blocks.

**cquanternary** The quanternary color, which only influences the foreground of example blocks.

```
701 \ifx\beamer@sjtubeamermin@color\beamer@sjtubeamermin@colorblue%
702   \definecolor{cprimary}{HTML}{004098}      %problue
703   \definecolor{csecondary}{HTML}{298626}     %lightgreen
704   \definecolor{ctertiary}{HTML}{004D4B}     %lightgray
705   \definecolor{cquanternary}{HTML}{FFFFFF}   %white
706 \else%
707   \definecolor{cprimary}{HTML}{9E1F36}      %engred
708   \definecolor{csecondary}{HTML}{F28101}    %orange
709   \definecolor{ctertiary}{HTML}{FED201}     %yellow
710   \definecolor{cquanternary}{HTML}{000000}   %black
711 \fi%
```

## 6.7 Logo

**\logo** Depend on the language definition, load the required logo by default. The logo is protected by the copyright from SJTU. The logo could be customized by redefinition from **\logo** command.

```
712 \logo{
713   \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langen
714     \includegraphics{sjtuenlogo.pdf}
715   \else
716     \includegraphics{sjtucnlogo.pdf}
717   \fi
718 }
```

### 6.7.1 Load TikZ

Load TikZ package and its related library: **pattern.meta** provides the interface to define a pattern; **fadings** provides the method to create a fading mask; **decoration.pathmorphing** provides the interface to user-define a decoration.

```
719 \RequirePackage{tikz}
720 \usetikzlibrary{patterns.meta}
721 \usetikzlibrary{fadings}
722 \usetikzlibrary{decorations.pathmorphing}
```

### 6.7.2 Shape Declarations

**stamp** Declare stamp pattern to make a stamp array.

The newest version of TikZ provides the interface to user-define a pattern. 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 TikZEdt.

```
723 \tikzdeclarepattern{
```

```

724 name=stamp,
725 parameters={
726   \pgfkeysvalueof{/pgf/pattern keys/size},
727   \pgfkeysvalueof{/pgf/pattern keys/xshift},
728   \pgfkeysvalueof{/pgf/pattern keys/yshift},
729 },
730 defaults={
731   size/.initial = 5pt,
732   xshift/.initial = 0pt,
733   yshift/.initial = 0pt,
734 },
735 bottom left={(
736   -0.5*\pgfkeysvalueof{/pgf/pattern keys/size}
737   +\pgfkeysvalueof{/pgf/pattern keys/xshift},
738   -0.4*\pgfkeysvalueof{/pgf/pattern keys/size}
739   +\pgfkeysvalueof{/pgf/pattern keys/yshift}
740 )},
741 top right={(
742   0.5*\pgfkeysvalueof{/pgf/pattern keys/size}
743   +\pgfkeysvalueof{/pgf/pattern keys/xshift},
744   0.4*\pgfkeysvalueof{/pgf/pattern keys/size}
745   +\pgfkeysvalueof{/pgf/pattern keys/yshift}
746 )},
747 tile size={(
748   \pgfkeysvalueof{/pgf/pattern keys/size},
749   0.8*\pgfkeysvalueof{/pgf/pattern keys/size}
750 )},
751 code={
752   \def\s{\pgfkeysvalueof{/pgf/pattern keys/size}}%
753   \tikzset{x=0.5*\s,y=0.2*\s}
754   \fill[xshift=\pgfkeysvalueof{/pgf/pattern keys/xshift},
755         yshift=\pgfkeysvalueof{/pgf/pattern keys/yshift}]
756     (-0.25*\s,0)
757     -- (-0.17*\s,0.06*\s)
758     -- (-0.17*\s,0.1*\s)
759     -- (0.17*\s,0.1*\s)
760     -- (0.17*\s,0.06*\s)
761     -- (0.25*\s,0)
762     -- (0.17*\s,-0.06*\s)
763     -- (0.17*\s,-0.1*\s)
764     -- (-0.17*\s,-0.1*\s)
765     -- (-0.17*\s,-0.06*\s) -- cycle;
766 }
767 }

```

`\stamparray` Create the stamp array in the TikZ environment.

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`.

```

768 \providecommand{\stamparray}[3]{
769   %#1: pattern size
770   %#2: starting point
771   %#3: ending point
772   \usebeamercolor{palette primary}
773   \fill [pattern={stamp[size=#1]},
774     pattern color=bg!50!fg] #2 rectangle #3;
775   \def\s{#1}%
776   \pgfmathparse{0.5*\s}\let\xs=\pgfmathresult%
777   \pgfmathparse{-0.4*\s}\let\ys=\pgfmathresult%
778   \fill [pattern={stamp[size=#1,xshift=\xs, yshift=\ys]},
779     pattern color=bg!50!fg] #2 rectangle #3;
780 }

```

**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.

```

781 \pgfdeclaredcoration{stampline}{initial}
782 {
783   \state{initial}[
784     width=\pgfdecorationsegmentlength,
785     auto corner on length=\pgfdecorationsegmentlength]
786   {
787     \def\l{\pgfdecorationsegmentlength}%
788     \pgfpathlineto{\pgfpoint{0.25*\l}{0pt}}
789     \pgfpathlineto{\pgfpoint{0.33*\l}{0.06*\l}}
790     \pgfpathlineto{\pgfpoint{0.33*\l}{0.1*\l}}
791     \pgfpathlineto{\pgfpoint{0.67*\l}{0.1*\l}}
792     \pgfpathlineto{\pgfpoint{0.67*\l}{0.06*\l}}
793     \pgfpathlineto{\pgfpoint{0.75*\l}{0pt}}
794     \pgfpathlineto{\pgfpoint{\l}{0pt}}
795   }
796   \state{final}
797   {
798     \pgfpathlineto{\pgfpointdecoratedpathlast}
799   }
800 }

```

Good Luck with SJTUBeamer MIN !

---

Developer

Log Creative

## 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, “L<sup>A</sup>T<sub>E</sub>X Sparkle Project.” [Online]. Available: <https://logcreative.github.io/LaTeXSparkle/>
- [5] TUG, “Upgrade from T<sub>E</sub>X 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/shxnnnjggqvp>
- [7] xu-cheng, “L<sup>A</sup>T<sub>E</sub>X-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 T<sub>E</sub>Xbook*. 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 T<sub>E</sub>X*, 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>

Copyright 2021 Log Creative & L<sup>A</sup>T<sub>E</sub>X Sparkle Project  
Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at  
<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The Current Maintainer of this work is Log Creative.