Developer's Guide on SJTUBeamer MIN

Log Creative

1.0 August 14, 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 Build

To make a CTAN package, a modern `l3build` method is adopted for building the package. And all souce 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 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.

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 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
```

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

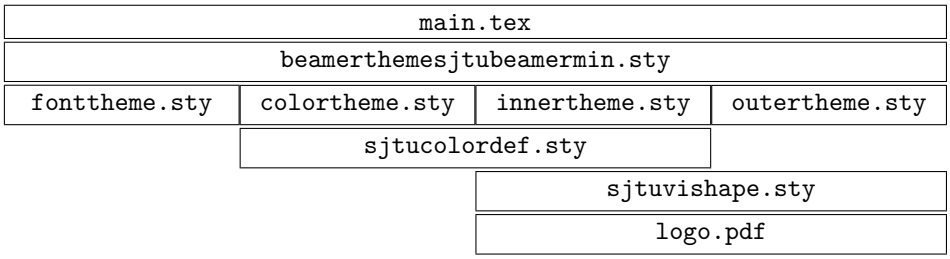
By the recommendation from beamer package[3], SJTUBeamer [MIN] uses the same modular architecture to build the template. Like it is in Java, to let the beamer template locate your theme, the style file has to be in the standard name.

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
lang
cjk
gbt
other settings

colortheme.sty	color
fonttheme.sty	
outertheme.sty	pattern,navigation,lang
innertheme.sty	pattern,color,lang

And this version meets the standingfree criteria. All source files could be used seperatly 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 quanternary	cquanternary	#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 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.

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 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.

5.2 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 Ti*k*Z (which is on top of PGF). You could almost draw any vectorized shapes by referring to Ti*k*Z documentation [12]. In short, Ti*k*Z 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, Ti*k*ZEdt 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-Seperated Values) respectively. As is mentioned, the author created a tool PGFPLOTSEDT to help such graphs in an 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 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 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 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 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.

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 Declartion

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

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

6.1.2 Beamer Color Interface

Load the common color library for sjtubeamermin.

```
\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.

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

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

This part defines the color scheme of title.

```
\setbeamercolor{background canvas}{bg=white}
\setbeamercolor{logo}{use={palette primary},bg=,fg=palette primary.fg}
\setbeamercolor{normal text}{fg=black,bg=black!40}
\setbeamercolor*{block title}{parent=structure}
\setbeamercolor{titlelike}{parent={palette primary}}
\setbeamercolor{title}{fg=cprimary,bg=}
\setbeamercolor{subtitle}{fg=csecondary,bg=}
```

This part defines the color of block title.

```
\setbeamercolor{block title}{fg=white,bg=cprimary!90}
\setbeamercolor{block title alerted}{use=alerted text,
  fg=white,bg=csecondary}
\setbeamercolor{block title example}{use=example text,
  fg=cquanternary,bg=ctertiary}
```

This part defines the color of block body.

```
\setbeamercolor{block body}{parent=normal text,use=block title,
  bg=block title.bg!30}
\setbeamercolor{block body alerted}{parent=normal text,
  use=block title alerted,bg=block title alerted.bg!30}
\setbeamercolor{block body example}{parent=normal text,
  use=block title example,bg=block title example.bg!30}
```

This part defines the color of footline.

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

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

```
\setbeamercolor{part title}{parent={palette primary}}
\setbeamercolor{section title}{parent={palette secondary}}
\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.

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

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

```
\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.

```
\usefonttheme{professionalfonts}
```

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

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

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

```
\setbeamerfont{part name}{size=\normalsize}
\setbeamerfont{section name}{size=\normalsize}
\setbeamerfont{subsection name}{size=\normalsize}
```

Set the font size in the footnote to footnotesize.

```
\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

```
\RequirePackage{pgfplots}
\RequirePackage{array}
\RequirePackage{colortbl}
\RequirePackage{booktabs}
\RequirePackage{pgfplotstable}
\RequirePackage{tcolorbox}
\RequirePackage{multicol}
```

6.3.2 Option Declaration

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

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

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

```
\DeclareOptionBeamer{pattern}{\def\beamer@sjtubeamermin@pattern{#1}}
\def\beamer@sjtubeamermin@patternnone{none}%
\def\beamer@sjtubeamermin@patterntitle{title}%
\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.

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

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

```
\ExecuteOptionsBeamer{
  lang=cn,
  color=blue,
  pattern=all
}
\ProcessOptionsBeamer
```

6.3.3 Shape Dependencies

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

```
\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.

```
\tikzfading[
  name=center fade,
  inner color=transparent!0,
  outer color=transparent!15
]
\tikzfading[
  name=fade right,
  left color=transparent!0,
  right color=transparent!100
]
```

Define the `title page` template.

```
\defbeamertemplate*{title page}{sjtubeamermin}[1][]
{
  \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.

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

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

```
  \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.

```
  \else%
    \ifx\beamer@sjtubeamermin@pattern\beamer@sjtubeamermin@patternnone%
    \else%
    \begin{tikzpicture}[overlay]
      \stamparray{20pt}
        {(-0.2*\the\paperwidth,-1*\the\paperheight)}
        {(1*\the\paperwidth, 0.2*\the\paperheight)}
      \fill [bg,path fading=fade right]
        (-0.2*\the\paperwidth,-1*\the\paperheight) rectangle
        (1*\the\paperwidth, 0.2*\the\paperheight);
      \fill [bg,path fading=center fade,xshift=-10pt,yshift=-20pt]
        (0.2*\the\paperwidth,0) circle [radius=\the\paperwidth];
    \end{tikzpicture}
    \fi%
  \fi%
```

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

```
  \vfill
  \begingroup
    \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 TEX code for typesetting. `\beamer@shortinstitute` meta is used to avoid compressing on `\par`, while `\insertinstitute` will force the input to spread on one signle line.

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

```
    \usebeamercolor{titlelike}
    \begin{beamercolorbox}{logo}
      \vskip8pt
      \hbox{
        \hskip4.5pt{\resizebox{!}{1cm}{\insertlogo}}
        \ifx\insertinstitute\@empty%
        \else
          \ifx\insertlogo\@empty%
          \else
            {\hskip3pt \vrule width0.5pt}\hskip7pt
          \fi
          \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
            \vbox{
              \fontsize{13pt}{0pt}\selectfont
              \insertinstitute
              \par\noindent\vskip0.15em
              \fontsize{5pt}{0pt}\selectfont
              \textsc{\insertshortinstitute}
              \baselineskip 3.2pt
              \par~
            }
          \else%
            \vbox to 1cm{
              \vfill
              \vbox{
                \offinterlineskip
                \noindent \strut
                \baselineskip 0pt \lineskip -2pt
                \scriptsize\textsc{\beamer@shortinstitute}
                \strut
              }
              \vfill
            }
          \fi%
        \fi%
      }
      \vskip8pt
    \end{beamercolorbox}
```

Insert title, subtitle, author, and date.

```
    \begin{beamercolorbox}[sep=8pt,#1]{title}
      \usebeamercolor[fg]{palette primary}
      \usebeamerfont{title}\inserttitle\par%
      \ifx\insertsubtitle\@empty%
      \else%
        \vskip0.25em%
        {\usebeamerfont{subtitle}\insertsubtitle\par}%
      \fi%
```

```
    \end{beamercolorbox}%
    \vskip1em\par
    \begin{beamercolorbox}[sep=8pt,#1]{author}
      \usebeamerfont{author}\insertauthor
    \end{beamercolorbox}
    \begin{beamercolorbox}[sep=8pt,#1]{date}
      \usebeamerfont{date}\insertdate
    \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.

```
    \begin{tikzpicture}[overlay,yshift=0.77em]
      \node [above left] at (0.88*\the\paperwidth,0)
        {\usebeamercolor[fg]{titlegraphic}\inserttitlegraphic};
    \end{tikzpicture}
  \endgroup
  \vskip0.5em
  \vfill
}
```

6.3.5 Part Page

Define the `part page` beamer template.

```
\defbeamertemplate*{part page}{sjtubeamermin}[1][]
{
  \vfill
  \vskip 8pt
  \begingroup
```

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

```
    \begin{beamercolorbox}[sep=16pt,right,#1]{part title}
      \hfill\usebeamerfont{part name}
      \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
       ~\insertromanpartnumber~
      \else%
        \partname~\insertromanpartnumber
      \fi%
      \par\vskip4pt
      \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.

```
      \hbox to \textwidth{
        \usebeamerfont{footline}%
        \setbeamercolor{temp}{fg=section in head/foot.fg}
        \setbeamercolor{section in head/foot}{use=palette primary,
          fg=palette primary.fg,bg=}
        \hfill
```

```
        \insertnavigation{0.4\textwidth}
        \hspace*{1cm}
        \setbeamercolor{section in head/foot}{fg=temp.fg}
      }
    \end{beamercolorbox}
  \endgroup
  \vfill
}
```

Redirect the part command to make a part page.

```
\AtBeginPart{
  \begin{frame}
    \partpage
  \end{frame}
}
```

6.3.6 Section Page and Subsection Page

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

```
\def\sectionblock#1{
  \begin{beamercolorbox}[sep=12pt,right,#1]{section title}
    \usebeamerfont{section name}
    \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
       ~\insertsectionnumber~
    \else%
      \sectionname~\insertsectionnumber
    \fi%
    \par\vskip4pt
    \usebeamerfont{section title}\insertsection\par
  \end{beamercolorbox}
}
```

Define the `section page` beamer template.

```
\defbeamertemplate*{section page}{sjtubeamermin}[1][]
{
  \vfill
  \begingroup
    \sectionblock{#1}
  \endgroup
  \vfill
}
```

Define the `subection page` beamer template.

```
\defbeamertemplate*{subsection page}{sjtubeamermin}[1][]
{
  \vfill
  \begingroup
    \sectionblock{#1}
    \begin{beamercolorbox}[sep=8pt,right,#1]{subsection title}
      \usebeamerfont{subsection name}
      \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn%
```

```
        ~\insertsubsectionnumber~
      \else%
        \subsectionname~\insertsubsectionnumber
      \fi%
      \par\vskip 4pt
      \usebeamerfont{subsection title}\insertsubsection\par
    \end{beamercolorbox}
  \endgroup
  \vfill
}
```

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.

```
\setbeamertemplate{items}[circle]
\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.

```
\RequirePackage{sjtucolordef}
```

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

```
\newtcbox{\highlight}[1][cprimary]{
    on line,
    arc=0pt,
    colback=#1,
    colupper=white,
    boxrule=0pt,
    boxsep=0pt,
    left=4pt,
    right=4pt,
    top=2pt,
    bottom=2pt
}
```

`\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.

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

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

```
\tcbuselibrary{skins}
\tcbuselibrary{listingsutf8}
```

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

```
\lstset{
  basicstyle=\ttfamily\small,
  keywordstyle=\color{cprimary},%
  stringstyle=\color{csecondary},%
  commentstyle=\color{ctertiary!80!gray},%
  columns=flexible,
  extendedchars=false,
  showstringspaces=false,
  showspaces=false,
  breaklines
}
```

`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.

```
\newtcblisting{codeblock}[2][]{
listing only,
listing engine=listings,
listing options={
  #1,%
  numbers=left,
  numberstyle=\color{cprimary!80}\ttfamily\scriptsize,
  numbersep=5pt,
},
enhanced,
sharp corners,
top=0mm,
bottom=0mm,
title=#2,
colback=cprimary!5,
colframe=cprimary!80,
overlay={
    \begin{tcbclipinterior}\fill[cprimary!20]%
        (frame.south west) rectangle ([xshift=5mm]frame.north west);
    \end{tcbclipinterior}
}
}
```

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.

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

```
  interior empty,
  sharp corners,
  top=2pt,bottom=2pt,left=2pt,right=2pt,
  borderline={4pt}{0pt}{
    #1,
    line width=0.5pt,
    decoration={
      stampline,
      segment length=8pt,
      path has corners=true,
    },
    decorate
  }
}
```

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.

```
\pgfplotsset{
  compat=newest,
  every axis/.style={
    font=\small\sffamily\sansmath,
    cycle multi list={
      mark options={thick}\nextlist
      mark list\nextlist
      cprimary,csecondary,ctertiary\nextlist
    },
    height=0.5*\the\paperheight,
    axis line style={
      cprimary,
      thin
    },
    every tick label/.style={
      cprimary,
      font=\small,
      /pgf/number format/assume math mode=true
    },
    grid style={cprimary!10},
    tick style={cprimary!50},
    minor tick style={cprimary!30},
    xlabel style={cprimary},
    ylabel style={cprimary},
    zlabel style={cprimary},
    legend style={
      draw={cprimary},
      thin,
      nodes={cprimary}
    },
    thick,
```

```
  },
}
```

6.3.10 Tables

Two macros are defined to make the header colored.

```
\def\zapcolorreset{\let\reset@color\relax\ignorespaces}
\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.

```
\pgfplotstableset{
  col sep=comma,
  every table/.style={
    font={\small\sffamily},
  },
  every head row/.style={
    before row={
      \arrayrulecolor{cprimary}
      \toprule
      \colorrows{\color{cprimary}}
    },
    after row={
      \midrule
      \colorrows{\color{black}}
    },
  },
  every last row/.style={
    after row=\bottomrule
  },
}
```

6.3.11 Footnotes

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

```
\defbeamertemplate*{footnote}{sjtubeamermin}
{
  \usebeamerfont{footnote}
  \parindent 0.5em\noindent%
  \raggedright
  \hbox to 1.5em{\hfil\insertfootnotemark}\insertfootnotetext\par%
}
```

6.3.12 Bottom Page

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

```
\def\bottompage{
```

Enter vertical mode.

```
\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.

```
\usebeamercolor{palette primary}
\usebeamercolor{palette secondary}
\begin{tikzpicture}[overlay,yshift=-80pt]
  \def\w{\the\paperwidth}%
  \def\h{\the\paperheight}%
  \fill [palette primary.bg] (-0.2*\w,-1*\h) rectangle (1*\w, 0.5*\h);
  \begin{scope}[fill=palette primary.bg!50!black,fill opacity=0.15]
    \clip (0.63*\w,1.05*\h) circle (1*\h);
    \fill (0.14*\w,-0.95*\h) circle (1.67*\h);
  \end{scope}
  \begin{scope}[fill=palette secondary.bg!50!palette primary.bg!70!white,
    fill opacity=0.15]
    \clip[xshift=26pt] (0.95*\w,-0.67*\h) circle (1.17*\h);
    \fill
      (0.14*\w,-0.95*\h) circle (1.67*\h)
      (0.63*\w,1.05*\h) circle (1*\h);
  \end{scope}
\end{tikzpicture}
```

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

```
\vfill
\begingroup
  \raggedleft
  \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`.

```
  \vfill
  \vskip6em
  \begin{beamercolorbox}[sep=8pt]{title}
    \usebeamercolor[fg]{palette primary}
    \usebeamerfont{title}\noindent
    \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langcn

    \else
      Thank You
    \fi
    \vskip1em%
    \usebeamerfont{subtitle}\insertauthor~$\cdot$~\inserttitle
  \end{beamercolorbox}%
  \vfill
  \vskip3.5em
\endgroup
}
```

`\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.

```
\def\makebottom{
  \ifbeamer@inframe%i
  \bottompage%
  \else%
  \frame{\bottompage}%
  \fi%
}
```

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.

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

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

```
\DeclareOptionBeamer{pattern}{\def\beamer@sjtubeamermin@pattern{#1}}
\def\beamer@sjtubeamermin@patternnone{none}%
\def\beamer@sjtubeamermin@patterntitle{title}%
\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.

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

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

```
\ExecuteOptionsBeamer{
  lang=cn,
  pattern=all,
  navigation=tools
}
\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.

```
\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 `\insernavigation` 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.

```
\ifx\beamer@sjtubeamermin@navigation\beamer@sjtubeamermin@navigationsubsections%
  \addtobeamertemplate{footline}{
    \vskip 4pt
    \vbox{}
    \ifnum\beamer@partstartpage=\c@page %
    \else
      \par\hfill\insertnavigation{0.4\paperwidth}\hspace*{0.1cm}
    \fi
    \par
    \vskip 10pt
    \vbox{}
  }{}
```

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

```
\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`.

```
  \def\pagenumbering{
    \raisebox{1.2pt}[0pt][0pt]{
      \usebeamerfont{footline}
      \usebeamercolor{footline}
      \color{footline.fg!50}
      \insertframenumber/\inserttotalframenumber
      \hspace*{0.5em}
    }
  }
```

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

```
  \addtobeamertemplate{navigation symbols}{}{%
    \hspace{1em}%
    \pagenumbering
  }%
```

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.

```
  \addtobeamertemplate{footline}{
    \ifnum\beamer@partstartpage=\c@page %
    \else%
      \hfill%
      \ifx\beamer@sjtubeamermin@navigation\beamer@sjtubeamermin@navigationtools%
        \usebeamertemplate***{navigation symbols}%
      \else
        \pagenumbering%
      \fi
    \fi%
    \hspace*{0.1cm}\par
    \vskip 4pt
  }{}
\fi%
```

6.4.3 Shape Dependencies

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

```
\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.

```
\tikzfading[
  name=fade left little,
  right color=transparent!0,
  left color=transparent!10
]
```

Define the beamer template `frametitle`. Some code is from the original beamer definition on `frametitle` (LPPL-1.3c).

```
\defbeamertemplate*{frametitle}{sjtubeamermin}[1][left]
{%
  \ifbeamercolorempty[bg]{frametitle}{}{\nointerlineskip}%
  \@tempdima=\textwidth%
  \advance\@tempdima by\beamer@leftmargin%
  \advance\@tempdima by\beamer@rightmargin%
  \begin{beamercolorbox}[sep=0.3cm,#1,wd=\the\@tempdima]{frametitle}
    \begingroup
    \usebeamerfont{frametitle}%
    \ifbeamer@draftmode%
    \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.

```
    \ifx\beamer@sjtubeamermin@pattern\beamer@sjtubeamermin@patternall
      \ifnum\beamer@sectionstartpage=\c@page %
        \begin{tikzpicture}[overlay]
          \ifx\insertframesubtitle\@empty%
            \def\h{-0.11*\the\paperheight}
          \else
            \def\h{-0.125*\the\paperheight}
          \fi
          \usebeamercolor{palette primary}
          \stamparray{20pt}
            {(-0.05*\the\paperwidth,\h)}
            {(\the\paperwidth,0.05*\the\paperheight)}
          \fill [bg,path fading=fade left little] (-0.05*\the\paperwidth,\h)
            rectangle (\the\paperwidth,0.05*\the\paperheight);
        \end{tikzpicture}
      \fi
    \fi
```

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

```
  \fi%
  \vbox{}
  \ifx\insertframesubtitle\@empty\vskip-2pt%
  \else\vskip-1ex\fi%
  \if@tempswa\else\csname beamer@fte#1\endcsname\fi%
  \strut\insertframetitle\strut\par%
  {%
    \ifx\insertframesubtitle\@empty%
    \else%
    {
      \usebeamerfont{framesubtitle}
      \usebeamercolor[fg]{framesubtitle}
      \strut\insertframesubtitle\strut\par
    }%
    \fi
  }%
  \vskip-1ex%
  \endgroup%
```

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

```
  \raggedleft%
  \begingroup
```

```
    \ifx\insertframesubtitle\@empty\vskip-2.5ex%
    \else\vskip-3.5ex\fi%
    {\resizebox{!}{0.7cm}{\insertlogo}}\hspace*{2ex}%
    \endgroup%
    \ifx\insertframesubtitle\@empty%
    \else\vskip0.5ex\fi%
    \if@tempswa\else\vskip-.3cm\fi%
  \end{beamercolorbox}%
}
```

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 avaliable for all modes.

```
\mode<all>
```

6.5.1 Option Declaration

`navigation` Change the appearence of the navigation bar, which will affect in the outer theme.

```
\DeclareOptionBeamer{navigation}{
    \PassOptionsToPackage{navigation=#1}{beamerouterthemesjtubeamermin}
}
```

`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.

```
\DeclareOptionBeamer{lang}{
    \def\beamer@sjtubeamermin@lang{#1}
    \PassOptionsToPackage{lang=#1}{beamerouterthemesjtubeamermin}
    \PassOptionsToPackage{lang=#1}{beamerinnerthemesjtubeamermin}
}
\def\beamer@sjtubeamermin@langcn{cn}%
\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}`.

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

`color` Provided two options:

blue The default selection.

red The recomended theme for non-scitific scenario.

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

```
\DeclareOptionBeamer{color}{
    \PassOptionsToPackage{color=#1}{beamercolorthemesjtubeamermin}
    \PassOptionsToPackage{color=#1}{beamerinnerthemesjtubeamermin}
}
```

`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.

```
\DeclareOptionBeamer{pattern}{
    \PassOptionsToPackage{pattern=#1}{beamerouterthemesjtubeamermin}
    \PassOptionsToPackage{pattern=#1}{beamerinnerthemesjtubeamermin}
}
```

`gbt` Choose the behaviour of citing.

false Use `biblatex` to cite.

bibtex Use `bibtex` to cite.

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

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

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

```
\ExecuteOptionsBeamer{
    navigation=tools,
    cjk=false,
    lang=cn,
    color=blue,
    pattern=title,
    gbt=false,
}
\ProcessOptionsBeamer
```

6.5.2 Option Execution

Disable the warning from `hyperref` which conflicts the setting in CTEX or CJK. It has to be manually disabled.

```
\RequirePackage{silence}
\def\Hy@WarnOptionDisabled#1{
    \def\next{#1}%
    \ifx\next pdfauthor %
        \ifx\next driverfallback %
        \else
        \Hy@Warning{%
            Option `#1' has already been used,\MessageBreak
            setting the option has no effect%
        }\fi%
    \fi%
}
```

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

```
\ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langen%
\else
    \ifx\beamer@sjtubeamermin@cjk\beamer@sjtubeamermin@cjktrue%
        \RequirePackage{CJKutf8}
        \renewcommand{\figurename}{ }
        \renewcommand{\tablename}{ }
        \renewcommand{\contentsname}{ }
    \else%
        \RequirePackage[UTF8]{ctex}
    \fi%
\fi
```

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

```
\ifx\beamer@sjtubeamermin@gbt\beamer@sjtubeamermin@gbtbibtex%
    \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langen%
        \bibliographystyle{IEEEtran}
    \else
        \RequirePackage{gbt7714}
    \fi
\else
    \ifx\beamer@sjtubeamermin@gbt\beamer@sjtubeamermin@gbttrue%
        \RequirePackage[style=gb7714-2015]{biblatex}
    \else
        \RequirePackage[style=authortitle-comp]{biblatex} %
    \fi
    \def\cite#1{
        \footfullcite{#1}
    }
\fi%
```

To avoid the messness of Chinese bookmarks.

```
\hypersetup{unicode}
\RequirePackage{bookmark}
\WarningFilter{latexfont}{Font shape}
```

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

```
\mode<presentation>
\beamer@compresstrue
\usecolortheme{sjtubeamermin}
\usefonttheme{sjtubeamermin}
\useoutertheme{sjtubeamermin}
\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.

```
\ifx\beamer@sjtubeamermin@color\beamer@sjtubeamermin@colorblue%
    \definecolor{cprimary}{HTML}{004098}        %problue
    \definecolor{csecondary}{HTML}{298626}      %lightgreen
    \definecolor{ctertiary}{HTML}{004D4B}       %lightgray
    \definecolor{cquanternary}{HTML}{FFFFFF}    %white
\else%
    \definecolor{cprimary}{HTML}{9E1F36}        %engred
    \definecolor{csecondary}{HTML}{F28101}      %orange
    \definecolor{ctertiary}{HTML}{FED201}       %yellow
    \definecolor{cquanternary}{HTML}{000000}    %black
\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.

```
\logo{
```

```
  \ifx\beamer@sjtubeamermin@lang\beamer@sjtubeamermin@langen
    \includegraphics{sjtuenlogo.pdf}
  \else
    \includegraphics{sjtucnlogo.pdf}
  \fi
}
```

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.

```
\RequirePackage{tikz}
\usetikzlibrary{patterns.meta}
\usetikzlibrary{fadings}
\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.

```
\tikzdeclarepattern{
  name=stamp,
  parameters={
    \pgfkeysvalueof{/pgf/pattern keys/size},
    \pgfkeysvalueof{/pgf/pattern keys/xshift},
    \pgfkeysvalueof{/pgf/pattern keys/yshift},
  },
  defaults={
    size/.initial = 5pt,
    xshift/.initial = 0pt,
    yshift/.initial = 0pt,
  },
  bottom left={(
    -0.5*\pgfkeysvalueof{/pgf/pattern keys/size}
      +\pgfkeysvalueof{/pgf/pattern keys/xshift},
    -0.4*\pgfkeysvalueof{/pgf/pattern keys/size}
      +\pgfkeysvalueof{/pgf/pattern keys/yshift}
  )},
  top right={(
    0.5*\pgfkeysvalueof{/pgf/pattern keys/size}
      +\pgfkeysvalueof{/pgf/pattern keys/xshift},
    0.4*\pgfkeysvalueof{/pgf/pattern keys/size}
      +\pgfkeysvalueof{/pgf/pattern keys/yshift}
  )},
  tile size={(
```

```
    \pgfkeysvalueof{/pgf/pattern keys/size},
    0.8*\pgfkeysvalueof{/pgf/pattern keys/size}
  )},
  code={
    \def\s{\pgfkeysvalueof{/pgf/pattern keys/size}}%
    \tikzset{x=0.5*\s,y=0.2*\s}
    \fill[xshift=\pgfkeysvalueof{/pgf/pattern keys/xshift},
      yshift=\pgfkeysvalueof{/pgf/pattern keys/yshift}]
      (-0.25*\s,0)
      -- (-0.17*\s,0.06*\s)
      -- (-0.17*\s,0.1*\s)
      -- (0.17*\s,0.1*\s)
      -- (0.17*\s,0.06*\s)
      -- (0.25*\s,0)
      -- (0.17*\s,-0.06*\s)
      -- (0.17*\s,-0.1*\s)
      -- (-0.17*\s,-0.1*\s)
      -- (-0.17*\s,-0.06*\s) -- cycle;
  }
}
```

`\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`.

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

`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.

```
\pgfdeclaredecoration{stampline}{initial}
{
  \state{initial}[
    width=\pgfdecorationsegmentlength,
    auto corner on length=\pgfdecorationsegmentlength]
  {
    \def\l{\pgfdecorationsegmentlength}%
```

```
    \pgfpathlineto{\pgfpoint{0.25*\l}{0pt}}
    \pgfpathlineto{\pgfpoint{0.33*\l}{0.06*\l}}
    \pgfpathlineto{\pgfpoint{0.33*\l}{0.1*\l}}
    \pgfpathlineto{\pgfpoint{0.67*\l}{0.1*\l}}
    \pgfpathlineto{\pgfpoint{0.67*\l}{0.06*\l}}
    \pgfpathlineto{\pgfpoint{0.75*\l}{0pt}}
    \pgfpathlineto{\pgfpoint{\l}{0pt}}
  }
  \state{final}
  {
    \pgfpathlineto{\pgfpointdecoratedpathlast}
  }
}
```

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, "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 Ti*k*Z," Oct. 2020. [Online]. Available: https://tuna.moe/event/2020/tikz/

[14] Thomas and Julian, "Ti*k*ZEdt – A semigraphical Ti*k*Z-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