

Moloch: a minimalist, feature-rich beamer theme

Johan Larsson, samcarter

Abstract

Moloch is a fork of the popular, but unmaintained, L^AT_EX beamer theme Metropolis. It fixes outstanding issues in Metropolis while preserving the original minimalist design. In this article we introduce Moloch’s features and options, provide guidance for Metropolis users transitioning to Moloch, and describe the documentation system that we use to generate both PDF and HTML documentation.

1 Introduction

The excellent Metropolis theme [?] provides a clean, professional look for L^AT_EX beamer [?] presentations as well as useful features such as progress bars and (what we think are) sane defaults for styling. The problem is that Metropolis is no longer actively maintained—its latest updates date six years back. Internally, it patches or redefines many internal macros and, with the evolution of beamer and the L^AT_EX kernel, these modifications have become brittle. Over time a number of issues have cropped up. Most of these are minor and corrigible through workarounds, but have caused users’ preambles to grow increasingly cluttered with such patches. Some users have also requested features, while others have offered contributions to help improve the theme. Metropolis being unmaintained, however, these requests and contributions have gone unaddressed.

This has led us to fork Metropolis into a new theme, which we call Moloch¹. The goal is to keep the original design philosophy and look of Metropolis while fixing outstanding issues and incorporating useful features and contributions from the community. We have modified the design slightly, but mostly with the intent of improving accessibility or to bring the theme closer to standard beamer behavior. We have also improved the internal code to make the theme robust and maintainable going forward. In particular, Moloch now uses the interfaces provided by beamer wherever possible, instead of patching internal macros. Users of Metropolis should find it easy to switch to Moloch, whilst at the same time benefit from the bug fixes, new features, and improved stability.

Since neither Metropolis nor Moloch have been introduced in the TUGboat before, we will begin by introducing Moloch in the next section. Users of Metropolis may want to skip ahead to Section 3,

which summarizes the changes made in Moloch and offers guidance for transitioning from Metropolis to Moloch. Finally, in Section 4, we describe the documentation system used for Moloch, which generates both PDF and HTML documentation from a single source.

2 Getting started with Moloch

Moloch is a standard beamer theme and is enabled in the usual way via `\usetheme{moloch}`. Here is a minimal example presentation using Moloch:

```

1 \documentclass{beamer}
2 \usetheme{moloch}
3
4 \molochcolors{normal text bg=white}
5
6 \title{Moloch Theme}
7 \subtitle{A Minimalist Beamer Theme}
8 \author{Johan Larsson\inst{1} \and samcarter}
9 \institute{
10   \inst{1} Department of Mathematical Sciences,
11   University of Copenhagen
12 }
13 \date{\today}
14 \titlegraphic{\includegraphics{images/logo}}
15
16 \begin{document}
17
18 \maketitle
19
20 \begin{frame}
21   \frametitle{A Sample Frame}
22   Hello, World!
23 \end{frame}
24
25 \end{document}
```

This produces the presentation shown in Figure 1. As you can see, Moloch provides a distinctive title slide with somewhat different formatting than the standard beamer title slide². Note that the default background color is a light gray (as in Metropolis); we have changed it to white here, using `\molochcolors` (see Section 2.1) to improve printability for TUGboat.

Unlike most themes, Moloch comes with a number of options that can be set either when loading the theme or via the macro `\molochset`. The benefit of using the latter is that you can change options midway through your presentations and limit them to the current scope. The following two calls are therefore two equivalent ways to modify the title page style.

```

1 \usetheme[titlepage=plain]{moloch}
2 \molochset{titlepage=plain}
```

¹ Perhaps familiar to you if you know your Metropolis.

² But if you prefer the default beamer one, then Moloch has the option to use that instead.

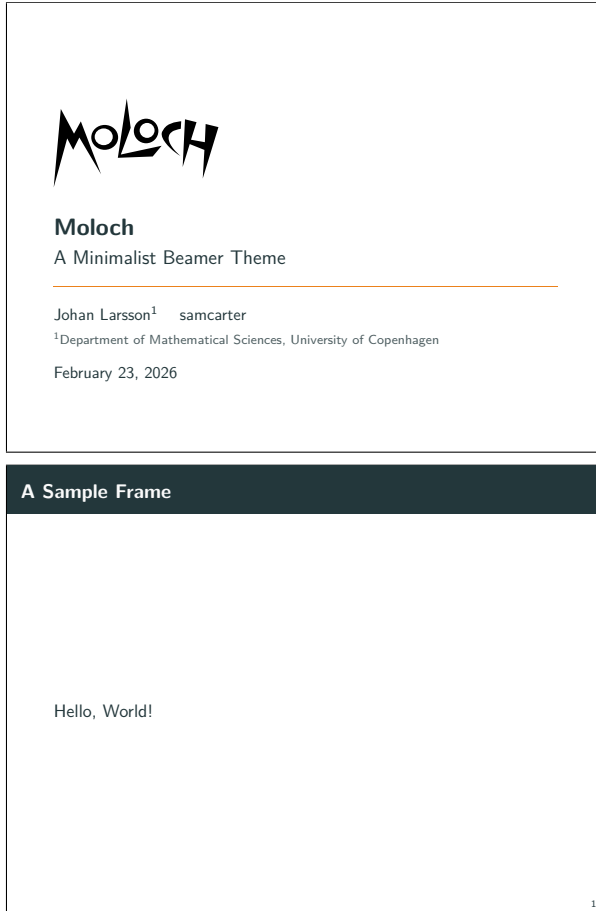


Figure 1: The title slide and first frame of a presentation made with Moloch.

Like many other `beamer` themes, `Moloch` is split into a color theme, font theme, and inner and outer themes. Here, we assume that the full `Moloch` theme is loaded via `\usetheme{moloch}`.

2.1 Color theming

A major new feature of `Moloch` is a full-fledged color theme system that allows users to customize the colors used in the theme in an intuitive way. It also supports the notion of light and dark modes, which can be switched between at any time in the presentation. In addition, we also support a number of pre-defined color themes.

To set the overall color theme, use the option `colortheme` and `colortheme variant` for selecting light or dark mode. For example, the following code snippet sets the color theme to `catppuccin` [?] in dark mode (Figure 2).

```
1 \molochset{
2   colortheme=catppuccin,
3   colortheme variant=dark
4 }
```

```
5
6 \begin{frame}
7   \frametitle{The Catppuccin Color Theme}
8
9   \begin{exampleblock}{This Frame}
10    This frame uses the \alert{catppuccin}
11    color theme in dark mode.
12   \end{exampleblock}
13
14   Colors rely on standard color definitions,
15   like
16   \begin{itemize}
17     \item \textcolor{mAlertFg}{
18       \texttt{mAlertFg}}
19     \item \textcolor{mExampleFg}{
20       \texttt{mExampleFg}}
21   \end{itemize}
22   which allows users to tap into the current
23   color theme's colors.
24 \end{frame}
```

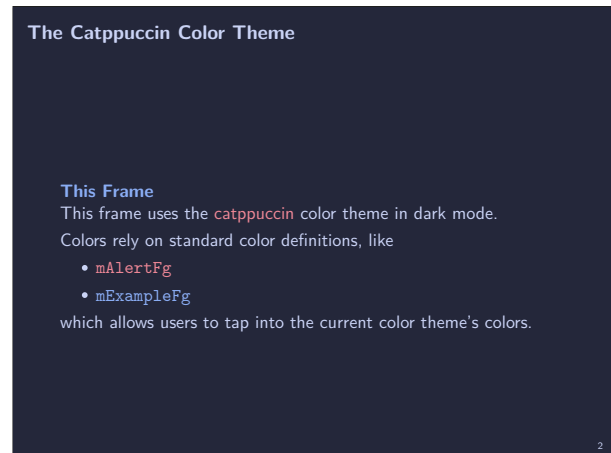


Figure 2: A frame using the `catppuccin` color theme in dark mode.

For more granular color customization, `Moloch` features a new macro, `\molochcolors`, which sets individual color elements of the theme. Users already familiar with `beamer`'s existing options for color customization (e.g. `\setbeamercolor`) might wonder what this macro brings to the table. The answer is that it allows users to separately specify colors for light and dark modes of the theme. The general syntax for this is

```
1 \molochcolors{<mode>/<element>=<color>}
```

where `<mode>` is either `light` or `dark`, `<element>` is the name of the color element to set, and `<color>` is the color specification.

For instance, the following code snippet customizes the frame title foreground and background colors, the progress bar color, and the alerted and example text colors in light mode (Figure 3).

```

1 \molochcolors{
2   light/frametitle fg=black,
3   light/frametitle bg=yellow!20,
4   light/progressbar fg=magenta!90,
5   light/alerted text=red!80!black,
6   light/example text=blue!60,
7 }

```

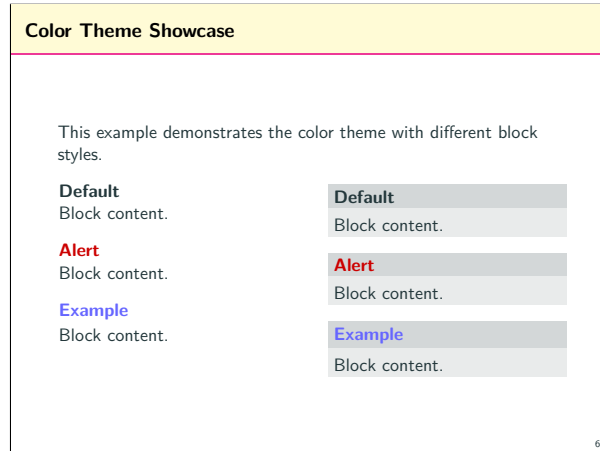


Figure 3: A frame demonstrating customized colors in light mode.

In addition, omitting the *mode* part will set the color for the current mode. For example, the following code snippet sets the color of example blocks to green in the current mode.

```

1 \molochcolors{
2   example fg=green!70!black
3 }

```

The end result is that **Moloch** provides a flexible and (we hope) intuitive color theming system that allows users to easily customize the look of their presentations and effortlessly switch between light and dark modes at their convenience. To enable this color system functionality, **Moloch** makes heavy use of the facilities of **pgfkeys** [?] from **PGF/TikZ** [?].

2.2 Progress bars

A popular feature of **Metropolis** (that **Moloch** retains) is the ability to track progress through the slides via a progress bar on either the frames or on section and/or subsection pages. To setup progress bars for frames, use the option **progressbar** with either **head**, **foot**, or **frametitle** to place the progress bar in the header, footer, or below the frame title, respectively. The following code snippet places a thick progress bar in the header of each frame (Figure 4).

```

1 \molochset{
2   progressbar=head,
3   progressbar linewidth=3pt
4 }

```

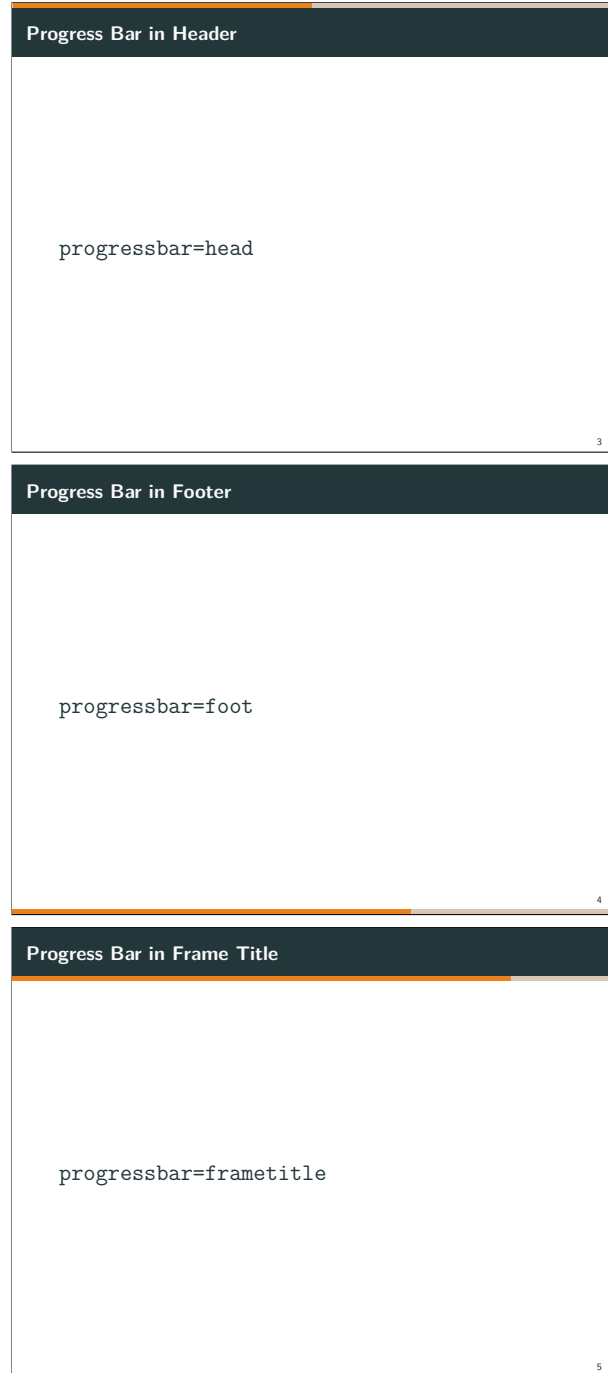


Figure 4: Progress bars in the footer, header, and below the frame title. Note that we have increased the line width of the bars to make them more visible. In the default setting, the bars are less obtrusive.

Unlike **beamer**, **Moloch** comes with section (but not subsection) pages enabled by default, which means that sections (using **\section**) results in dedicated frames. By default, they use the same progress bar style as frames (Figure 5), but this can be changed with **\molochset{sectionpage=<option>}**.

The available options are **progressbar** (the default), **none** (no section page) and **simple** (a section page without a progress bar).

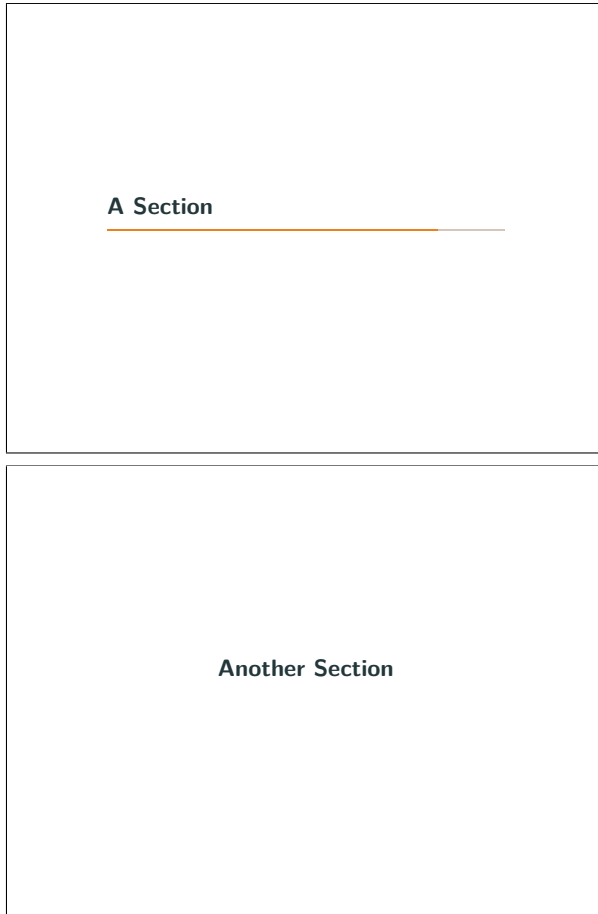


Figure 5: The default (left) and simple (right) section pages styles. The section page can also be disabled entirely via `\molochset{sectionpage=none}`.

2.3 Standout frames

Moloch (as well as **Metropolis**) supports special *standout* frames, which are intended as an alternative style for highlighting important frames in the presentation. They are created by specifying a **standout** key to the **frame** environment. For example, the following code snippet creates the standout frame seen in Figure 6, which in the default theme results in a frame with a dark background and a centered, larger text.

```
1 \begin{frame}[standout]
2   Thank you!
3 \end{frame}
```

As with most elements of **Moloch**, the colors of standout frames can be customized via the color theme system.

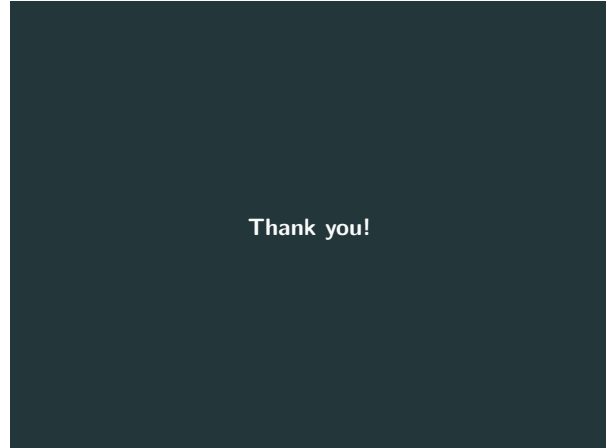


Figure 6: A standout frame created via `\begin{frame}[standout] <content> \end{frame}`.

2.4 Title page styles

As we have already seen, **Moloch** provides a distinctive title page style by default. As usual, however, preferences differ and for that reason, **Moloch** comes with three different title page styles that can be selected via the **titlepage** option. The available styles are **moloch** (the default), **plain** (a centered, **beamer**-like style), and **split** which splits the information by a vertical line (Figure 7).

3 Transitioning from Metropolis to Moloch

Moloch retains much of the original design and behavior of **Metropolis**, but there *are* differences. In this section we summarize these, explain their motivation, and help users transition from **Metropolis** to **Moloch**. Although many of the changes in **Moloch** are internal code improvements and mitigations of bugs, we will not focus on these here and instead highlight changes that will be visible to users.

One important difference is that we have tried to make **Moloch** more consistent with standard **beamer** behavior, which makes it easier for **beamer** users to adopt **Moloch**—at the cost of requiring **Metropolis** users to adapt. A guiding principle has been to not re-invent the wheel when **beamer** already provides the desired functionality. This has led to some changes in behavior compared to **Metropolis**, as we now, for instance, prefer to rely on **beamer**’s built-in mechanisms for frame numbering and definitions of block environments. Another related change is to avoid redefining internal **beamer** macros. This was a source of brittleness in **Metropolis** that forced upstream workarounds in **beamer** to avoid breaking changes. By relying on standard **beamer** mechanisms instead, **Moloch** is more robust and future-proof.

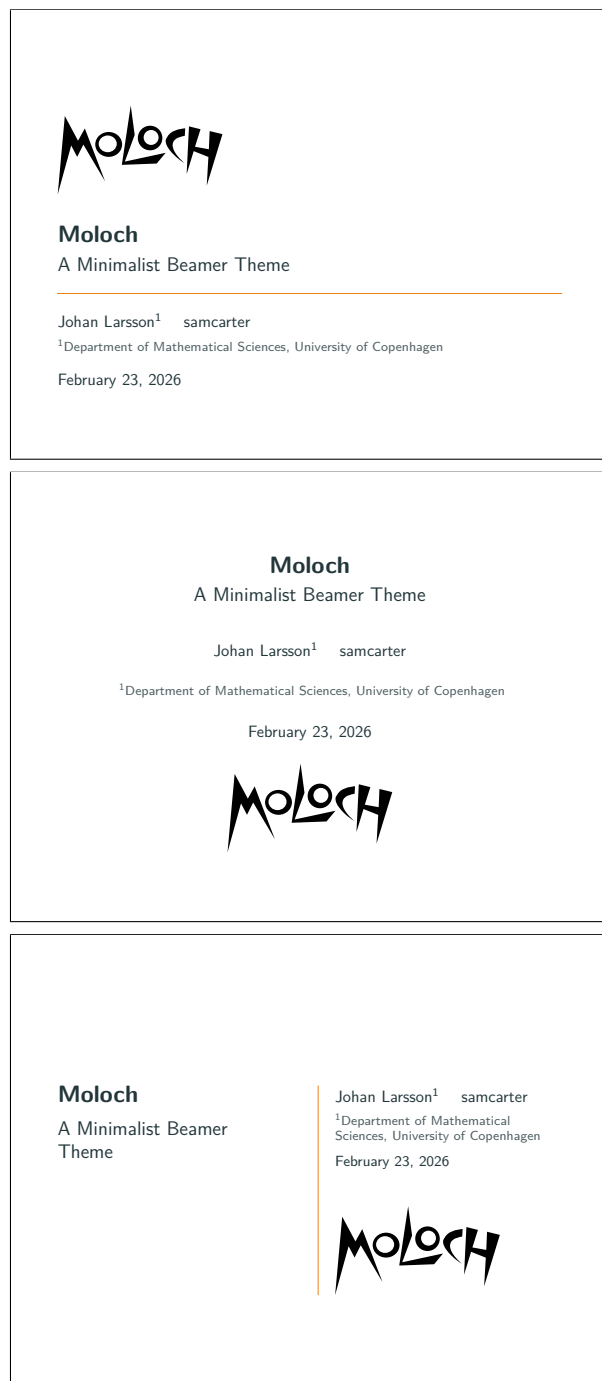


Figure 7: The three title page styles: moloch (left), plain (center), and split (right).

3.1 Typography

Metropolis features a preference for **Fira Sans** (designed by Erik Spiekermann). For users compiling with $\text{Lua}\text{\LaTeX}$ and $\text{X}\text{\LaTeX}$, the package automatically tries to find, configure, and use this font. We appreciate the design of **Fira Sans**, but also feel that the font choice should be left to the user, particularly since the behavior in **Metropolis** leads different users to see different fonts depending on their system setup.

In **Moloch**, we have therefore removed the automatic font selection and instead leave the choice of font to the user. Those who still wish to use **Fira Sans** can do so by adding the following lines to their preamble.

```
1 \usepackage[medium,light]{FiraSans}
2 \usepackage{FiraMono}
3 \ifTUTeX
4   \usepackage{firamath-otf}
5 \fi
```

To better mimic the original **Metropolis** setup, options `weight=Light` and `usefilenames` can be passed to `firamath-otf`, but note that this requires manual installation of the appropriate weights of the **Fira Math** font³.

Users of $\text{Lua}\text{\LaTeX}$ and $\text{X}\text{\LaTeX}$ may instead wish to directly use `fontspec` [?] to set up **Fira Sans** as the main font. Here is an example configuration that matches the original **Metropolis** setup.

```
1 \usepackage{fontspec}
2 \usepackage{unicode-math}
3
4 \setsansfont[
5   ItalicFont={Fira Sans Light Italic},
6   BoldFont={Fira Sans},
7   BoldItalicFont={Fira Sans Italic}
8 ]{Fira Sans Light}
9
10 \setmonofont[
11   BoldFont={Fira Mono Medium}
12 ]{Fira Mono}
13
14 \setmathfont[
15   Scale=MatchLowercase,
16   ItalicFont={Fira Math Italic},
17   BoldFont={Fira Math Bold}
18 ]{Fira Math}
19
20 \AtBeginEnvironment{tabular}{%
21   \addfontfeature{Numbers={Monospaced}}
22 }
```

³ A full set of weights is available from github.com/firamath/firamath.

3.2 Setting options

Metropolis introduces `\metroset` for setting theme options. **Moloch** instead uses `\molochset`, which is otherwise used exactly in the same way.

3.3 Paragraph spacing and line spacing

Unlike standard **beamer**, in which `\parskip` is set to zero, **Metropolis** instead sets it to 0.5em units. This means that in **Metropolis**, you do not need to use manual paragraph spacing commands like `\medskip` to separate paragraphs.

But this had some undesired side-effects, such as introducing additional spacing inside block environments, around captions, and list environments. It also meant that users could not easily switch from **Metropolis** to standard **beamer** without having to adjust their content to accommodate the different paragraph spacing. For those reasons, we have removed this setting in **Moloch**. That means that users transitioning will need to either separate paragraphs manually (e.g. via `\medskip`) or use `\setlength{\parskip}{0.5em}` to reinstate the original behavior.

Metropolis also increases line spacing compared to standard **beamer**, which we have also removed in **Moloch** for similar reasons and to improve consistency with standard **beamer**. Users who wish to reinstate this behavior can do so via `\linespread{1.15}`.

Note, however, that changing the paragraph or line spacing may affect the overall layout of the slides and you may need to adjust other elements accordingly.

3.4 New example color

While we generally find **Metropolis** well-designed, we believe that the color used in example blocks can be difficult to read for some users, especially those with color vision deficiencies, as well as print poorly in grayscale. Therefore, we have changed from the lighter and vivid green (■) in **Metropolis** to a darker, but less saturated teal (■) in **Moloch**. Users who prefer the original color can revert via the following snippet.

```
1 \definecolor{metropolisGreen}{HTML}{14B03D}
2 \molochcolors{example fg=metropolisGreen}
```

3.5 Frame subtitles

Moloch, unlike **Metropolis**, supports frame subtitles. The reasoning for this omission was that the author felt they were unnecessary clutter. We generally agree that subtitles should be used sparingly (if at all), but we also do not want to force this preference upon our users.

3.6 Frame numbering

Metropolis sports its own frame-numbering system, which requires a separate package⁴ to have frame numbering restart (and not count towards the total number) for appendix slides. **beamer** now supports this natively, so **Moloch** simply uses this built-in functionality. The design is *slightly* different, with smaller font size and adjusted margins.

3.7 Block environments

Metropolis supports an alternative filled block style, which uses a different background color for block environments. Many other **beamer** themes support filled blocks too. Unlike most of these, however, **Metropolis** alters the alignment of the text in these filled block environments such that the main body text (for the frame) aligns with the left edge of the box and not the text inside the box. Users have, however, reported problems with this approach. And because we prefer the default behavior in **beamer** also for typographical reasons and consistency⁵, we have reverted to the standard **beamer** behavior in **Moloch**. See Figure 8 for a comparison.

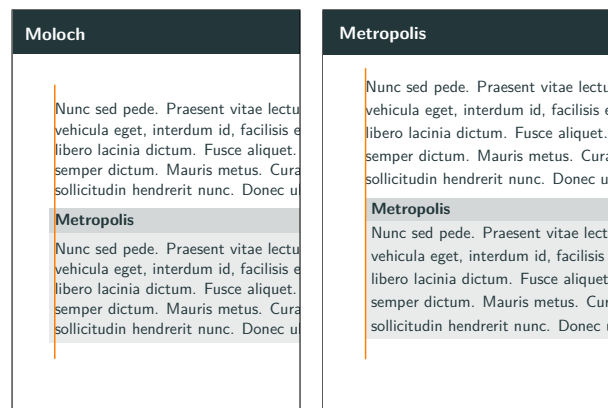


Figure 8: A comparison of alignment of block environments in **Moloch** (left) and **Metropolis** (right).

3.8 Itemize lists

Metropolis uses `\textbullet` (●) as the symbol for all levels of the itemize environment. The problem with `\textbullet` is that it is defined differently in different fonts. Notably, for instance, in *Latin Modern* [?] it is defined as a filled square, which looks quite different from the round bullet used in most other fonts.

The appearance of itemize symbols is a frequent source of issues because they typically depend on

⁴ `appendixnumberbeamer` [?]

⁵ Since changing the block style in **Metropolis** also risks modifying the layout of the slide.

the font. Another common setting is to use a math symbol, like $\bullet$ ($\bullet$), but then the problem is only shifted to depend on the math font instead.

For these reasons, **Moloch** uses PGF primitives to draw all of its itemize symbols, which means that they work regardless of the font used. Also, to improve the visual hierarchy, **Moloch** uses different symbols for different levels of the itemize environment (Figure 9). Users who nevertheless wish to switch to the **Metropolis** symbol can simply specify the following in their preamble.

```
1 \setbeamertheme{itemize items}{\textbullet}
```

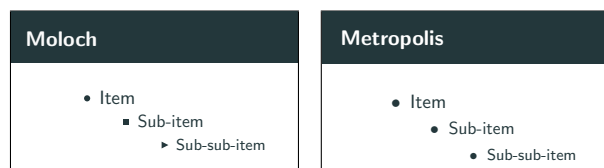


Figure 9: The design of itemize lists in **Moloch** (left) and **Metropolis** (right).

3.9 PGFPlots theme

Metropolis includes a custom **pgfplots** [?] theme that uses colors from Paul Tol’s 21-color theme [?]. We removed this theme from **Moloch** since we felt that it was outside the scope of a **beamer** theme to provide a plotting theme. The theme is still provided through **Metropolis**, however, so users who wish to use it can do so by adding the following line to their preamble.

```
1 \usepackage{pgfplots,pgfplotsthemetol}
```

3.10 Other miscellaneous changes

Moloch includes many other changes. We will not go into all the details here; but briefly, we have

- redesigned the default title page style,
- allowed customizing beamer templates in the appendix,
- optimized section pages for presentations with wider aspect ratios,
- removed hard-coding of the bibliography style, and
- modernized the build system to use **l3build** [?] and include unit tests.

4 Documentation

L^AT_EX packages typically come with PDF documentation and **Moloch** is no exception, featuring both a user guide and implementation documentation. Yet, although it is standard to supply documentation as a PDF manual, this format has some downsides. Notably, it is not responsive to screen size — being hard

to read and navigate on small screens and making suboptimal use of the screen size on large, widescreen displays. It also lacks good search functionality, is difficult to cross-reference outside of the document itself, and, perhaps most importantly, is less accessible than web documentation.

Moloch therefore also comes with web documentation available at moloch.innk. The site features HTML-based versions of both the user guide and implementation documentation, in similar fashion as the unofficial documentation for PGF/TikZ at tikz.dev. Yet, whereas that site is generated via **l^warp** [?], **Moloch** takes a somewhat different approach. Since we think this approach might be of interest to other package authors, we will briefly summarize it here.

4.1 Architecture

The manual for the package is in fact written in **Markdown**, specifically the flavor of **Quarto** [?], which is built on **Pandoc** [?]. **Quarto** has powerful features for turning **Markdown** source into both PDF books and HTML web sites. This allows us to provide a single source of truth for both the PDF and HTML documentation.

One challenge is that we prefer the literate programming style of **docstrip** [?] and therefore document the implementation using standard **L^AT_EX**, which means we maintain documentation in both **Markdown** and **L^AT_EX**. Fortunately, **Pandoc** supports conversion both to *and* from **L^AT_EX**, which allows us to convert the documentation from the **.dtx** format to **Markdown** after stripping it through a custom script.

This then gives us a complete **Markdown** source file for the entire documentation, which we then convert to both a PDF manual (through **L^AT_EX**) and an HTML website via **Quarto**. In order to provide proper styling for documentation of macros and package options, we use **Lua** filters, which is a built-in feature of **Quarto**, to keep a unified syntax for the documentation. Here is, for instance, an example of a documentation block for the **block** option in **Moloch**.

```
1 ::: {.describe-option
2   option-name="block"
3   values="transparent, fill"
4   default="transparent"
5 }
6 Optionally adds a light grey background to block
7 environments like `theorem` and `example`.
8 :::
```

The output of this block is shown in Figure 10 both as rendered in **L^AT_EX** and HTML.

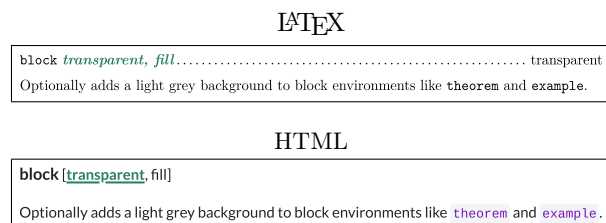


Figure 10: The same documentation block rendered in L^AT_EX (top) and HTML (bottom) via a custom Quarto filter.

4.2 Reusability

Our documentation system is part of the internal machinery of *Moloch* and would currently require some effort to adapt to another package. But we hope that this section may have piqued some interest and would be happy to collaborate to turn this into a reusable feature — perhaps as a separate package — in order to help other authors provide web documentation for their packages.

5 Conclusion

We have presented *Moloch*, a fork of the popular *beamer* theme *Metropolis*. *Moloch* retains the minimalist design philosophy of *Metropolis* while fixing outstanding issues and incorporating new features. Our hope is that *Moloch* will serve as a robust and maintainable alternative to *Metropolis* going forward and that users of both “regular” *beamer* themes and *Metropolis* will find it easy to adopt.

Moloch is developed at github.com/jolars/moloch and its documentation is available at moloch.ink. We welcome contributions and feedback from the community to help improve *Moloch* going forward.

6 Acknowledgments

We would like to first thank Matthias Vogelgesang for creating *Metropolis*. Without his work, *Moloch* would not exist. We would also like to thank all the users of *Metropolis* who have reported issues, requested features, and contributed code over the years. Many of these contributions have found their way into *Moloch*. And last but not least, we would like to thank all the users of *Moloch*. Their feedback and suggestions facilitate further improvement of the theme.

References

- [1] J. Allaire, C. Teague, et al. *Quarto*, Sept. 2025. doi.org/10.5281/zenodo.5960048
- [2] P. Carl. The Paul Tol 21-color salute. tradeblotter.com, Feb. 2013. tradeblotter.com.

- wordpress.com/2013/02/28/the-paul-tol-21-color-salute/
- [3] B. Dunn. The *lwarp* package, version 0.920, 2025-12-10. ctan.org/pkg/lwarp
- [4] C. Feuersänger, H. Menke, et al. The *pgf* package, version 3.1.11a, 2025-08-29. ctan.org/pkg/pgf
- [5] C. Feuersänger, H. Menke, The PGF/TikZ Team. The *pgfplots* package, version 1.18.2, 2025-08-14. pgfplots.sourceforge.net/
- [6] B. Jackowski, J.M. Nowacki. The *lm* package, version 2.005, 2021. www.gust.org.pl/projects/e-foundry/latin-modern
- [7] J. Lelong. The *appendixnumberbeamer* package, version 1.2, 2018. github.com/jlelong/appendixnumberbeamer
- [8] J. MacFarlane, A. Krewinkel, J. Rosenthal. *Pandoc*, May 2025. Version 3.7.0.2. github.com/jgm/pandoc
- [9] F. Mittelbach, W. Robertson, L^AT_EX3 team. *l3build* — A modern Lua test suite for T_EX programming. *TUGboat* 35(3):287–293, 2014. tug.org/TUGboat/tb35-3/tb111mitt-l3build.pdf
- [10] F. Mittelbach, The LaTeX Project Team. The *docstrip* package, version 2.6c, 2025-09-23. ctan.org/pkg/docstrip
- [11] samcarter, J. Wright, et al. The *beamer* package, version 3.76, 2025-08-13. ctan.org/pkg/beamer
- [12] T. Tantau. The *pgfkeys* package, version 2007-10-08, 2007-10-08. ctan.org/pkg/pgfkeys
- [13] The Catppuccin organization. *Catppuccin*. catppuccin.com, 2026.
- [14] The LaTeX Project Team, W. Robertson. The *fontspec* package, version 2.9g, 2025-09-29. ctan.org/pkg/fontspec
- [15] M. Vogelgesang. The *beamertheme-metropolis* package, version 1.2, 2017-01-23. ctan.org/pkg/beamertheme-metropolis

◇ Johan Larsson
Department of Mathematical
Sciences, University of
Copenhagen
Universitetsparken 5
2100 Copenhagen
Denmark
jola@math.ku.dk
jolars.co
ORCID 0000-0002-4029-5945

◇ samcarter