

Retour d'expérience sur le développement en \LaTeX 3 de Gotham

GUTenberg exposé mensuel

Romain NOËL¹²
romainoel@free.fr

¹Université Gustave Eiffel

²Inria

6 février 2025



Qui suis-je?

- utilisateur \LaTeX depuis 13 ans et de $\text{\LaTeX}3$ depuis 3 ans
- depuis j'ai travaillé dans 5 différentes institutions, adaptant régulièrement les templates en \LaTeX
- aujourd'hui, chercheur en physique computationnelle à l'Univ. Eiffel & Inria
- collaboration régulière avec autres instituts donc adaptation des templates...
- expérimenté, mais pas expert



1. Pourquoi ?
2. Gotham Theme
3. L'envers du décors
4. Et la suite ?

1. Pourquoi ?

Il est une fois

Pour mes différents templates, j'utilisais le très bon **Metropolis Beamer**.

L'adaptation pouvait être un peu longue pourtant une fois faite, tout le monde était content.



Figure 1: Exemples de présentations adaptées de **Metropolis**

Chaque changement avait tendance à casser les réglages fins (et donc la mise en page).

Un certain nombre de correctifs devait être appliqué en plus de **Metropolis**.

Même si ce n'a pas été exprimé explicitement, **Metropolis** semble ne plus être maintenu.

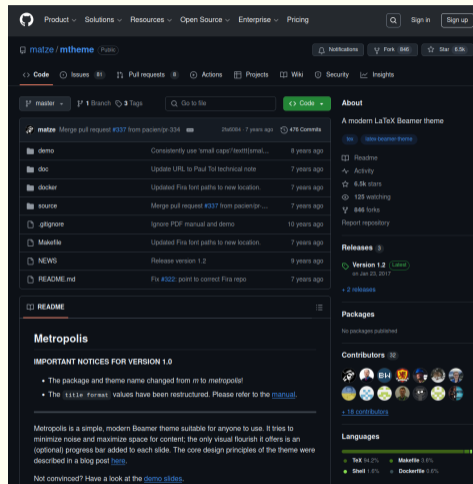


Figure 2: Aperçu de Metropolis sur Github

Mon idée était de créer une nouvelle version de **Metropolis** corrigeant les bugs résiduels, offrant plus de flexibilité pour s'adapter facilement aux templates (tordus ?) institutionnels.

Ceci dans l'espoir de passer d'un template à un autre en une commande, sans souci ni réglage. Ainsi je pourrai appliquer un principe fondamental faisant toute la beauté de \LaTeX : **séparer le fond de la forme.**

2. Gotham Theme

Gotham package

The **GOTHAM** theme is a Beamer theme with a minimal-ish visual style largely inspired by the METROPOLIS Beamer Theme by Matthias VOGELGESANG (and some other Beamer themes).

Yet, **GOTHAM** is highly extendable and versatile.

First, enable the theme by classically loading it:

```
\documentclass{beamer}  
\usetheme{gotham}
```

Then, all the customization can be performed at any moment in the presentation using:

```
\gothamset{<option>=...}
```

2. Gotham Theme

2.1. Fonts

Gotham title formats

Note, that you have to have Mozilla's *Fira Sans* font and XeTeX or LuaTeX installed to enjoy this wonderful typography.

GOTHAM supports 4 different title formats

```
\gothamset{format frametitle=}
```

- regular
- lower
- UPPER
- Title case

GOTHAM supports 3 different title shape

```
\gothamset{shape frametitle=...}:
```

- regular
- SMALL CAPS
- *italic*

They can either be set at once for every title type or individually.

This frame uses the title format options: `format frametitle=upper,`
`shape frametitle=italic.`

This frame uses the title format options: `shape frametitle=smallcaps`,
`format frametitle=titlecase`.

Potential Problems

Be aware that not every font supports small caps. If for example you typeset your presentation with pdfTeX and the Computer Modern Sans Serif font, every text in `smallcaps` will be typeset with the Computer Modern Serif font instead. Please refer to the documentation if you consider using it.

As a rule of thumb: just use it for plaintext-only titles.

This frame uses the title format options: `format frametitle=lower,`
`shape frametitle=regular.`

Presentation style via background color

The color mode (a.k.a. background color) can be changed using:

```
\gothamset{background=dark | light | transparent}
```


Blocks

Three different block environments are pre-defined and may be styled with an optional background color.

```
\gothamset{  
  block=native}
```

Default

Block content.

Alert

Block content.

Example

Block content.

```
\gothamset{  
  block=transparent}
```

Default

Block content.

Alert

Block content.

Example

Block content.

```
\gothamset{  
  block=fill}
```

Default

Block content.

Alert

Block content.

Example

Block content.

Color customization

The color theme can be used only in preamble with `\usecolortheme{wolverine}` and without guarantees on the visual aspect.

GOTHAM offers predefined color setup at any time through `\gothamset{colorset=red}`

Otherwise, the colors can be changed manually using:

```
\colorlet{colorPale}{gPaleYell} % BG in light/normal mode
\colorlet{colorDark}{gDarkBlack} % FG in light/normal mode
\colorlet{colorA}{gDarkTeal} % frametitle, stdin.out,
\colorlet{colorAreversed}{gLightTeal} % frametitle, stdin.in,
\colorlet{colorB}{gMidGrey} % gray BG : progress bar, blocks
\colorlet{colorC}{gDeepYellOr} % progress bar
\colorlet{colorD}{gLightOrange} % alert
\colorlet{colorE}{gLightGreen} % example
```

2. Gotham Theme

2.3. Inner

GOTHAM offers the possibility to adapt the title page layout (printed with `\maketitle` or `\titlepage`). This can be achieved using:

```
\defbeamertemplate{title page}{your name}{your defintion}  
\gothamset{title page= your name}
```

GOTHAM also predefined several templates such as: `gotham normal` | `gotham splitvert` | `gotham dividedpic` | `gotham reversed`

Table of contents

GOTHAM comes with the possibility to apply different styles for your table of contents (ToC) page. You can define your own ToC style as it follows:

```
\defbeamertemplate{toc page}{your name}{your def}  
\gothamset{tocframe template= your name}
```

Then, referring to this template using the frame option `[toc]` in your presentation:

```
\begin{frame}[toc]{Table of contents}  
  \tableofcontents%[hideallsubsections]  
\end{frame }
```

Or using one of the **GOTHAM** predefined templates, such as:

```
gotham simple | gotham bullet
```

Sections

GOTHAM provides a multiple options to tune sections (respectively `part`, `section`, `subsection` and `subsubsection`).

The section command `\section{Elements}` from Beamer will appear very different. The section page will appear or disappear thanks to:

`\gothamset{sectionframe default=<on|off>}`, while its layout (when appearing) is controlled by:

```
\defbeamertemplate{part|sub|subsub|section frame}
  {your name}{your def}
\gothamset{sectionframe template= your name}
```

GOTHAM predefined template are: `gotham progressbar` | `gotham simple` | `gotham splitvert progressbar` | `gotham splitvert simple` | `gotham progressvert`

After the section page, you can (de)activate a page with a table of contents for the section using `\gothamset{sectiontocframe default=<on|off>}`, and its layout is controlled by:

```
\defbeamertemplate{toc subsection frame}{your name}{your def}  
\gothamset{sectionframe template= your name}
```

GOTHAM predefined template are: `gotham simple` | `gotham bullet`

With **GOTHAM** you can locally or globally add watermark to your slides by using:

```
\defbeamertemplate{background}{watermark/your name}{your def}  
\gothamset{watermark template= your name}
```

Then, this watermark can be turned on locally using `\begin{frame}[watermark]` or globally with `\gothamset{watermark default= on}`.

Standin

GOTHAM comes with 2 environments/special layouts named `standin` and `standout`. These special layouts can be used to emphasize some content or last slide...

This layout can be turned on using `\begin{frame}[standin]` or using the dedicated environment (`\begin{standinenv}\begin{frame}...\end{frame}\end{standinenv}`).

Note that the background can also be tuned using:

```
\defbeamertemplate{background canvas}{standin/name}{your def}
\gothamset{standin BG template= name}
```

Standout

Here is an example of standout (working as standin), which can be combined with a watermark.

Another difference, apart the obvious color change is the font size and series.

2. Gotham Theme

2.4. Outer

GOTHAM defines a custom Beamer template to add a text to the footer. It can be set via

```
\setbeamertemplate{frame footer}{My custom footer}
```

Even after redefining (or not) your frame footer template, you can locally remove it with the frame option `\begin{frame}[nofooter]`.

rotateFooter

The default footer from **GOTHAM**, it displays the `shortdate`, `shorttitle` and `shortauthor`. So by filling these fields in your document setup, you will see them appear in your footer:

```
\title[your shorttitle]{Your title}  
\date[shortdate]{\today}  
\author[your shortauthor name]{John DOE}
```

Since we always need some extra space on some frames that would like to overlay a bit the footer, **GOTHAM**'s footer also offers possibility to be put locally on the side using `\begin{frame}[rotateFooter]`, or globally with

```
\gothamset{rotateFooter default=on}
```

If it has set globally, it can be deactivated locally with the frame option

```
\begin{frame}[norotateFooter].
```

GOTHAM has two hook commands, `\gothamRightFiligrane` and `\gothamLeftFiligrane`, that can be redefined to customize what to display in the edgings (a.k.a. filigrane, a.k.a. sidebar). As an example, one could do:

```
\renewcommand{\gothamRightFiligrane}{%  
  \rotatebox{90}{gotham right filigrane pattern}  
}
```

Then, to set if it should be displayed or not, globally

```
\gothamset{edging default=on}
```

or locally with the frame option `\begin{frame}[edging]` or `\begin{frame}[noedging]`.

Really wide contents

If you want a really wide content in your frame, you can change the size of your margin (requires `\usepackage{changepage}` in your preamble). You can also suppress the edging (`[noedging]`) and footer (`[nofooter]`) or even more radically footline (`[nofootline]`).

Here is an example combining them:

```
\begin{frame}[noedging,nofootline]{extended frame}
  \begin{adjustwidth}{-2em}{-2em}% 2em extra to the left and 2em for right margin.
    wide content
  \end{adjustwidth}
\end{frame }
```



The frametitle template brought by **GOTHAM** is relatively classic: it supports `\subframetitle` and frame continuation (with `[allowframebreaks]`) through templates that can be tuned. Nevertheless, it the frametitle template also includes a hook for your institute logo in the top right corner, leaving the command `\logo{}` free for your extra logos.

So, one can have both logos using:

```
\renewcommand{\gothamInstituteLogoSquare}[1][4ex]{
  \includegraphics[height=#1]{gotham-logo.pdf}
}
\logo{extra LOGO}
```


Numbering and progressbar

GOTHAM theme can numbering your frames in the bottom right corner using different styles. You can also decide to use a progression bar to indicate how much of your presentation remains.

The setup of numbering and progression bar can be performed through:

```
\gothamset{numbering= totalframenumber, progressbar position=foot}
```

Numbering available options are:

```
none | framenumber | totalframenumber | appendixframenumber | pagenumber  
| totalpagenumber | circle
```

Progressbar position available options are:

```
none | head | frametitle | foot | circlehead | left | right
```

3. L'envers du décors

L'idée était d'essayer de faire un thème en utilisant L^AT_EX3 (le premier ?), afin de ne pas faire une nouvelle bifurcation du thème `Metropolis` Beamer telle que `Moloch` ou `Trigon`.

Ce choix pour L^AT_EX3 était au début à la fois un goût personnel, une volonté de moderniser les thèmes Beamer et se démarquer.

À l'usage ce choix c'est montré utile pour pouvoir rendre `GOTHAM` plus flexible. Par exemple, le changement des éléments contexte tels que les `background` ou les `sidebar` dans des diapos de section `sectionframe`.

GOTHAM propose beaucoup de combinaison d'option (trop ?), ce qui implique un risque important de bugs. \implies Il me semblait utile de pouvoir tester indépendamment les différentes fonctions développées, puis leurs bonnes interactions.

GOTHAM a ainsi été écrit avec une méthode de *Test Driven Development* (TDD).

Et il utilise un système d'intégration continu/déploiement continu (CI/CD) afin de s'assurer de la non-régression du code et de pas diffuser une version (trop) buggée. Le CI/CD est assuré par un serveur Gitlab qui fait une batterie teste en environnement contrôlé a chaque changement avant d'autoriser un déploiement.

Pour mettre en place un TDD, il est nécessaire d'avoir un système de test adapté à \LaTeX . À ma connaissance, `l3build` est un des rares outils le faire.

`l3build`, bien que pas toujours très intuitif, permet en plus des tests, d'extraire les commentaires du code, de générer des exemples, une documentation et de packager le code source pour son déploiement.

Néanmoins, il peut être assez déroutant qu'un tel outil ne soit pas capable d'être un *build system* pour des documents classiques.

- colorset
- frametitle
- footer
- toc
- sectionpage
- standout
- ...



Figure 3: Page de garde et ToC Inria

- colorset
- sidebar
- toc
- sectionpage
- standout
- ...

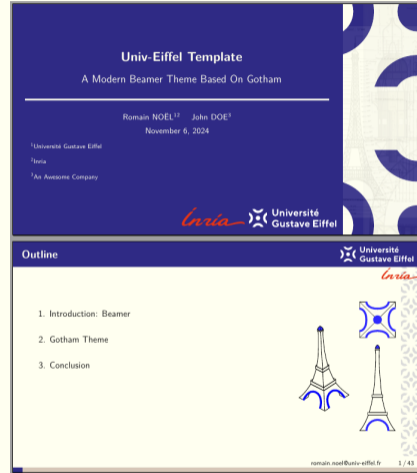


Figure 4: Page de garde et ToC Univ. Eiffel

Exemple 3/3

- colorset
- frametitle
- footer
- sectionpage
- standout
- ...



The screenshot shows two slides from a Beamer presentation. The top slide is the title page, featuring the logos for CEA, ISAS, and Université Gustave Eiffel. The title is "Inria Template" and the subtitle is "A Modern Beamer Theme Based On Gotham Long Long Long Long Long Long Long Long Long Long Long Long Long Long Long Long Long Long". The date is "November 6, 2024". The author information is "Romain NOEL^{1,2} John DOE³". The bottom slide is the table of contents, titled "Outline", listing the following sections: 1. Introduction: Beamer, 2. Gotham Theme (with sub-sections 2.1. Fonts, 2.2. Colors, 2.3. Inner, 2.4. Outer), and 3. Conclusion. The footer of the second slide includes the CEA logo, the text "INRIA/ISAS, 2024", "Inria Template", the email "romain.noel@inria.fr", and "1 / 43".

Figure 5: Page de garde et ToC CEA

4. Et la suite ?



- améliorer la vitesse de compilation: peut-être en découpant en sous thèmes certains `\defbeamertemplate`
- corriger les bugs
- ajouter des thèmes de couleurs
- uniformiser certains processus internes et variables

En développant **GOTHAM** et contribuant au package **derivative**, je voulais séparer les problèmes liés à l'environnement **l3build** de ceux liés au code lui-même.

Pour ce faire, j'ai créé un squelette de package \LaTeX 3 utilisant **l3build**.

Depuis, j'ai intégré mon REX dans ce squelette et j'ai ajouté l'outil **cookiecutter** afin pouvoir générer une base de package fonctionnelle.

https://gitlab.com/RomainNOEL/latex3_template_pkg

Notamment, j'ai entrepris un nouveau couple de packages pour séparer le fond de la forme pour l'écriture des tenseurs et des opérations tensoriels.

Par exemple, un vecteur peut s'écrire: \vec{v} ou \underline{v} ou \mathbf{v} ou encore v_α .

Et un gradient de vecteur est un tenseur pouvant s'écrire: $\text{grad}(\mathbf{v})$ ou $\vec{\nabla}(\vec{v})$ ou $\partial_\alpha(v_\beta)$...

Alors qu'il s'agit des mêmes objets (le fond), seule la forme change et cette dernière peut être fastidieuse à écrire.

D'où, un prototype de package: <https://gitlab.com/RomainNOEL/tensors>.

5. Résultat des comptes

GOTHAM permet :

- intégrer \LaTeX dans vos présentations
- utilise `13build` pour code TDD avec CI/CD
- une grand flexibilité pour adapter vos templates
- de basculer d'un template à l'autre à la volée

GOTHAM est disponible :

- CTAN: <https://ctan.org/pkg/beamertheme-gotham>
- Gitlab: <https://gitlab.com/RomainNOEL/beamertheme-gotham>
- exemples des adaptations:
https://github.com/Universite-Gustave-Eiffel/beamer-template_uge-inria

Toute aide est la bienvenue !

Merci pour votre attention !

—

Avez-vous des questions ?