

GOTHAM

A modern, minimal-ish, versatile and extendable theme
for Beamer

Romain NOEL

romainoel@free.fr

v1.2.1.c — 2025-02-11

Contents

1	Introduction	2
2	Getting Started	3
2.1	Installing from CTAN	3
2.2	Installing from Gitlab	3
2.3	A Minimal Example	4
2.4	Dependencies	4
3	Customization	5
3.1	Package options	5
3.1.1	Font theme	6
3.1.2	Color theme	6
3.1.3	Inner theme	7
3.1.4	Outer theme	9
3.2	Setup all the options	13
3.3	Frame options	14
3.4	Commands for Customization	15

3.5	Color Customization	19
3.6	Font Customization	19
3.6.1	Old style figures	19
3.7	Length Customization	20
4	Tips & Tricks	22
4.1	Backup Slides	22
4.2	Sources of inspiration	22
5	Known Issues	23
5.1	Title formats	23
5.2	Interactions with other color themes	24
5.3	Notes on second screen	24
5.4	Standout frames with labels	25
5.5	Standout frames with Pandoc	25
5.6	Other issues	25
6	License	26

1 Introduction

Beamer is an awesome way to make presentations with \LaTeX , but its theme selection is surprisingly sparse. The stock themes share an aesthetic that can be a little cluttered, while the few distinctive custom themes available are often specialized for a particular corporate or institutional brand.

The goal of **Gotham** is to provide a simple, modern Beamer theme suitable for anyone to use. It tries to minimize noise and maximize space for content; some visual flourishes are offered as options.

The final product provides a modern, elegant and versatile theme with a high degree of customization. So it can easily be used to build a particular corporate or institutional brand theme on top of it.

The implementation of this theme is strongly inspired from the excellent **METROPOLIS** theme by Matthias Vogelgesang, from which it borrows numerous options. **Gotham**'s codebase lives on [Gitlab](#).

If you have issues, find mistakes in the manual or want to help make the theme even better, please get in touch there. Any feature request, issue report or merge requests are welcome.

2 Getting Started

2.1 Installing from CTAN

The latest stable version of **Gotham** is available on [CTAN](#) and should now be part of the usual \TeX distributions (\TeX Live, \MacTeX , \MikTeX), under the name *beamertheme-gotham*. It means that if your distribution is kept up-to-date, the package should normally be already installed on your system. If this is not the case, consider updating the packages of your \TeX distribution.

For \TeX Live and \MacTeX users, this usually means running

```
tlmgr update --all
```

or if administrative privileges are required

```
sudo tlmgr update --all
```

For \MikTeX users, please refer to [the official \$\text{\MikTeX}\$ documentation](#).

2.2 Installing from Gitlab

If you want to use the cutting-edge development version of **Gotham**, you can install it manually by following these steps:

Download the source from [Gotham repository](#) using `git clone` or as a [zip archive](#) of the latest development version.

Compile the style files by running `l3build unpack` inside the downloaded directory. (Or run \LaTeX directly on `source/gothamtheme.ins`.)

Move the resulting *.sty files to the folder containing your presentation. To use **Gotham** with many presentations, run `l3build install` or move the *.sty files to a folder in your \TeX path instead.

Use the theme by declaring `\usetheme{gotham}` in the preamble of your document.

Gotham uses the `l3build` system to offer the following installation options for advanced users:

`l3build unpack` builds the theme style files.

`l3build doc` builds this documentation manual and the examples.

`l3build check` builds the theme and manual.

`l3build clean` removes the files generated by `l3build`.

`l3build install` installs the theme into your local texmf folder.

`l3build uninstall` removes the theme from your local texmf folder.

2.3 A Minimal Example

The following code shows a minimal example of a Beamer presentation using **Gotham**, and the output generated is given on fig. 1.

```
\documentclass{beamer}
\usetheme{gotham} % Loads Gotham theme
\title{A minimal example}
\subtitle{using Gotham theme}
\date{\today}
\author{John DOE}
\institute{Your university or company}

\begin{document}
  \maketitle
  \section{First Section}
  \begin{frame}{First Frame}
    Hello world !
  \end{frame}
\end{document}
```

2.4 Dependencies

Gotham depends on the `beamer` class and the following standard packages:

- `tikz`
- `expl3`
- `xfp`
- `etoolbox`
- `ifxetex`
- `ifluatex`



Figure 1: A simple example.

For best results, we recommend installing the fonts **Fira Sans** and **Fira Mono** and compiling with **Gotham** using XeLaTeX or LuaTeX. These are optional dependencies; **Gotham** is compatible with (e.g.) pdfL^AT_EX and will fall back to standard fonts if **Fira Sans** or **Fira Mono** is not installed.

The packaged name of **Fira Sans** is **Fira Sans OT** in some Linux distributions; this case is automatically handled by **Gotham**.

3 Customization

3.1 Package options

Gotham provides a number of options, which can be set using a key=value interface. The primary way to set options is to provide a comma-separated list of option-value pairs when loading **Gotham** in the preamble:

```
\setheme[option1=value1, option2=value2, ...]{gotham}
```

Options can be changed at any time — even mid-presentation ! — with the `\gothamset` command.

```
\gothamset{option1=newvalue1, option2=newvalue2, ...}
```

The list of options is structured as shown in the following example.

option key *list of possible values*..... default

A short description of the option.

As **Gotham** implementation is heavily inspired from the excellent **METROPOLIS** theme by Matthias Vogelgesang, many of METROPOLIS options are also available in **Gotham**.

3.1.1 Font theme

format title	<i>regular, lower, upper, titlecase</i>	<u>regular</u>
format subtitle	Individually controls the format/case of titles, subtitles, frametitle, framesubtitles, part, section and subsection titles.	
format frametitle		
format framesubtitle		
format part		
format section		
format subsection		
shape title	<i>regular, italic, smallcaps</i>	<u>regular</u>
shape subtitle	Individually controls the shape/series of titles, subtitles, frametitle, framesubtitles, part, section and subsection titles.	
shape frametitle		
shape framesubtitle		
shape part		
shape section		
shape subsection		

FRAMETITLE OF A FRAME
Subtitle frame

(a) Example
format frametitle=titlecase,
shape frametitle=smallcaps,
format framesubtitle= lower,
shape framesubtitle=regular.

frametitle of a frame
Subtitle frame

of
(b) Example of format frametitle=lower,
shape frametitle=italic,
format framesubtitle= regular,
shape framesubtitle=italic.

Figure 2: Examples of format and shape settings.

3.1.2 Color theme

The included **Gotham** color theme is used by default, but its colors can be easily changed to suit your tastes. All of the theme's styles are defined in terms of a few main colors:

- **colorA** The primary theme color, used for frametitle, standout and text if the appropriate options are selected.

An easy way to customize the theme is to redefine these colors using:

```

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

```

background *light, dark, transparent*..... transparent

Controls weather the color of all headings (title page, frame title, etc.) should be in black (default) or in a slightly darker shade of the theme color `theme`.

colorset *red, anthracite*..... anthracite

Predefined set colors (colorA, colorB, ...) leading to different ambiances.

block *native, transparent, fill*..... native

Optionally adds a light grey background to block environments like `theorem` and `example`.

3.1.3 Inner theme

title page *gotham normal, gotham reversed, gotham dividedpic, gotham splitvert, <your-name>*..... gotham normal

By setting this option you can change the title page according predefined style or even set your own template. If you want to use your own template, this latter should be previously defined with: `\defbeamertemplate{title page}{<your-name>}{<your-defintion>}`

.

watermark default *on, off*..... off

Enable or disable the watermark background template by default (ie. without using `\begin{frame}[watermark]`).

watermark template *gotham draft, <your-name>*..... gotham draft

Set the watermark background template to use when the `watermark` option is activated (by default or locally). If you want to use your own template, this latter should be previously defined with: `\defbeamertemplate{background}{watermark/<your-name>}{<your-defintion>}`.

standin template *gotham, <your-name>*..... gotham

standout template

added=2024-11-11

Set the standin and stantout template to use when the `standin` or `stantout` frame option is activated. If you want to use your own template, this latter should be previously defined with: `\defbeamertemplate{standout}{<your-name>}{<your-defintion>}`.

standin BG template *gotham, <your-name>*..... gotham

standout BG template

updated=2024-11-11

Since most of the time the standin and standout are only varying from their backgrounds, **Gotham** offers the possibility to change only and simply these local background through this option. This option sets the standin and stantout background template to use when the `standin` or `stantout` frame option is activated. If you want to use your own template, this latter should be previously defined with: `\defbeamertemplate{background}{standin/<your-name>}{<your-defintion>}`.

partframe default *on, off*..... on

sectionframe default

subsectionframe default

subsubsectionframe default

Enable or disable the display of the part frame (section, subsection and subsubsection respectively) at each part (other respectively) increment.

partframe template *gotham progressbar, gotham simple, gotham splitvert progressbar, gotham*

sectionframe template

subsectionframe template

subsubsectionframe template

splitvert simple, gotham progressvert..... gotham progressbar
Set the frame template to use when the `part` (or `section`, `subsection`, `subsubsection` respectively) frame option is activated (ie. using `\begin{frame}[part]`). If you want to use your own template, before giving its name to this option, your template should be defined with: `\defbeamertemplate{part frame}{<your-name>}{<your-defintion>}`.

tocframe template *gotham simple, gotham bullet, <your-name>*..... gotham bullet

Set the table of contents template to use when the `toc` option is activated. If you want to use your own template, this latter should be previously defined with:

```
\defbeamertemplate{part in toc}{<your-name>}{<your-defintion
>}
\defbeamertemplate{section in toc}{<your-name>}{<your-
defintion>}
\defbeamertemplate{subsection in toc shaded}{<your-name>}{<
your-defintion>}
\defbeamertemplate{subsubsection in toc shaded}{<your-name
>}{<your-defintion>}
\defbeamertemplate{background canvas}{toc/<your-name>}{<your-
definition>}
\defbeamertemplate{toc page}{<your-name>}{<your-defintion ,
set them now by exmaple>}
```

parttocframe default *on, off*..... on

sectiontocframe default Enable or disable the display of the table of content frame after the part frame (section,
subsubsection and subsubsection respectively) at each part (other respectively) increment.
subsubsection in toc shaded

parttocframe template *gotham simple, gotham bullet, <your-name>*..... gotham bullet

sectiontocframe template Set the frame template to use when the table of contents at each new part (or section,
subsubsection and subsubsection respectively). This new frame is internally using the
subsubsection in toc shaded `tocpart` frame option to activate the frame template. If you want to use your own
template, before giving its name to this option, your template should be defined with:
`\defbeamertemplate{part frame}{<your-name>}{<your-defintion>}`.

3.1.4 Outer theme

sidebar canvas right template *gotham filigrane, empty, <your-name>*..... gotham filigrane

sidebar canvas left template Setting templates for left and right sidebar canvas which are activated when giving the
frame option `\begin{frame}[edging]`. If you want to use your own template, before giving
its name to this option, your template should be defined with: `\defbeamertemplate`
`{sidebar canvas right}{default/<your-name>}{<your-defintion>}`

edging default *on, off*..... off

The option `edging default=on` can enable the `sidebar canvas right` (and `sidebar canvas left`) templates on every frame; but it can still be turned off for specific frames when using the frame option `\begin{frame}[noedging]`.

frametitle template *gotham subsameline, gotham subnewline, <your-name>*.. gotham subsameline

Option to set the frametitle template. **Gotham** offers one template that adds the subtitle on the same line (`gotham~subsameline`) and one that adds the subtitle on a new line (`gotham~subnewline`). If you want to use your own template, before giving its name to this option, your template should be defined with: `\defbeamertemplate{frametitle}{default/<your-name>}{<your-defintion>}`

framesubtitle template *gotham subnewline, <your-name>*..... gotham subnewline

Setting the template to use when the subtitle is on a new line. If you want to use your own template, before giving its name to this option, your template should be defined with: `\defbeamertemplate{framesubtitle}{default/<your-name>}{<your-defintion>}`

frametitle continuation *gotham, beamer, tot, <your-name>*..... gotham
template

Setting the template that is used in the frametitle when a frame too long and is continued/broke into several frames (using the frame option `allowframebreaks`). If you want to use your own template, before giving its name to this option, your template should be defined with: `\defbeamertemplate{frametitle continuation}{default/<your-name>}{<your-defintion>}`

numbering *none, framenummer, totalframenummer, appendixframenummer, pagenummer, totalpagenummer, circle*..... none

Setting the template with the frame number at the bottom right of each slide.

footer template *gotham, <your-name>*..... gotham

Setting the template that appears in the footer of the frame. `gotham footer` print the `shortdate` at right, the `shorttitle` in the middle and the `shortauthor` at left. Since in 16/9 the height is precious, `gotham` template also offers the possibility to rotate `shortdate` and `shortauthor` so they appear on sides. If you want to use your own template, before giving its name to this option, your template should be defined with: `\defbeamertemplate{frame footer}{default/<your-name>}{<your-defintion>}`

rotateFooter default *on, off*..... off

Enable or disable the `rotateFooter` frame option by default (ie. without using `\begin{frame}[rotateFooter]` on every frame). If the option is activated, it can be deactivated locally using the frame option `\begin{frame}[noRotateFooter]`.

mini frames shape *default (bullet from beamer), tick, box, gotham minibullet, gotham box, gotham minibox, <your-name>* gotham minibullet

Setting the shape of the mini frames to use, if the mini frame bundle refers to it (which is usually the case). If you want to use your own template, before giving its name to this option, your template should be defined with:

```
\defbeamertemplate{mini frame}{<your-name>}{
  <your-defintion>
}[action]{
  \setbeamersize{mini frame size=.1cm, mini frame offset=.05
    cm}
}
\defbeamertemplate{mini frame in current subsection}{<your-
  name>}{
  <your-defintion>
}
```

mini frames bundle *gotham mini, beamer, gotham nano <your-name>*..... gotham mini

Setting the set of symbols used in the mini frame navigation. If you want to use your own template, before giving its name to this option, your template should be defined with:

```
\defbeamertemplate{miniframe~home}{<your-name>}{<your-def>}
\defbeamertemplate{miniframe~current~slide}{<your-name>}{<
  your-def>}
\defbeamertemplate{miniframe~done~current~section}{<your-name
  >}{<your-def>}
\defbeamertemplate{miniframe~todo~current~section}{<your-name
  >}{<your-def>}
\defbeamertemplate{miniframe~done~other~section}{<your-name
  >}{<your-def>}
\defbeamertemplate{miniframe~todo~other~section}{<your-name
  >}{<your-def>}
\defbeamertemplate{miniframe~part}{<your-name>}{<your-def>}
\defbeamertemplate{miniframe~section~current}{<your-name>}{<
  your-def>}
\defbeamertemplate{miniframe~section~done}{<your-name>}{<your
  -def>}
\defbeamertemplate{miniframe~section~todo}{<your-name>}{<your
  -def>}
\defbeamertemplate{miniframe~subsection~current}{<your-name
  >}{<your-def>}
\defbeamertemplate{miniframe~subsection~todo}{<your-name>}{<
  your-def>}
\defbeamertemplate{miniframe~subsection~done}{<your-name>}{<
  your-def>}
```

mini frames compress *on, off*..... on

A shortcut for the beamer option `compress` that affects the mini frames.

mini frames nav spreading *spreading, centering, left, right*..... spreading

Controls how the mini frame should spread in the navigation bar.

mini frames nav sectioning *none, section, secsubsection*..... none

Setting the section in head/foot template that is used on top of the mini frame navigation bar.

mini frames nav position *none, head, foot, left, right*..... none

Setting where the mini frame navigation bar should appear.

progressbar position *none, head, frametitle, foot, circlehead, left, right*..... none

Setting where the progressbar should appear. The different positions are pretty obvious from their name, except for `circlehead`. This latter option is putting a circular progressbar around the frametitle-logo (from frametitle template defined with gotham theme). It worth noting that by doing so, the frametitle is using the command `\gothamInstituteLogoCircle` instead of `\gothamInstituteLogoSquare` which is used otherwise.

progressbar style *rectangle, rounded box, moving circle, fixed circle*..... rectangle

Setting how the progressbar should appear. `rectangle` option is using sharp rectangle for the progressbar, while `rounded box` is using rounded box and adds the percentage of progression at its right. `moving circle` and `fixed circle` are concerning the option `progressbar position=circle` only. `moving circle` add the number of the frame in a circle moving (following) the progression of the bar, while `fixed circle` put this framenummer constantly at the right of the circular progressbar.

progressbar advancement *tlbr, brlt*..... tlbr

Defines in which direction the progressbar should evolve. `tlbr` is the shortcut for: starting from the top left corner, it goes bottom then right, ie. anti-clockwise. `brtl` is the shortcut for: starting from the bottom right corner, it goes top then left, ie. clockwise.

progressbar mfn *on, off*..... off

Enable or disable the display of the mini frame navigation bar within the progressbar.

3.2 Setup all the options

```
\gothamset{
  % from font theme
  format title, shape title,
  format subtitle, shape subtitle,
  format frametitle, shape frametitle,
  format framesubtitle, shape framesubtitle,
  format part, shape part,
  format section, shape section,
  format subsection, shape subsection,
  % from color theme
  background,
```

```

block,
colorset,
% from inner theme
title page,
watermark template, watermark default,
standout template, standin template,
standout BG template, standin BG template,
partframe template, partframe default,
sectionframe template, sectionframe default,
subsectionframe template, subsectionframe default,
subsubsectionframe template, subsubsectionframe default,
tocframe template,
parttocframe template, parttocframe default,
sectiontocframe template, sectiontocframe default,
subsectiontocframe template, subsectiontocframe default,
%subsubsectiontocframe template, subsubsectiontocframe
    default,
% from outer theme
sidebar canvas right template, sidebar canvas left
    template,
edging default,
frametitle template, framesubtitle template, frametitle
    continuation template,
numbering,
rotateFooter default,
footer template,
mini frames shape, mini frames bundle, mini frames
    compress, mini frames nav spreading,
progressbar position, progressbar style, progressbar
    advancement, progressbar mfn,
}

```

3.3 Frame options

Frame options are affecting the templates used on the current frame through the following syntax:

```
\begin{frame}[option] ... \end{frame}
```

Below a description of the different frame options brought by **Gotham**.

`noBGC` Apply an empty background canvas template.

`watermark` Apply (with `watermark`) or deactivated (`nowatermark` when `\gothamset{watermark default=on}` was given) the background template, through `\defbeamertemplate{background}{watermark/<your-name>}{<your-def>}` and `\gothamset{watermark template=<your-name>}`.

`standout` Apply the `standin` (and `standout` respectively) templates through the definition `\standin defbeamertemplate{standin}{<your-name>}{<your-def>}` and the option `\gothamset{standin BG template=<your-name>}`.

`toc` Apply the `toc` template defined by `\gothamset{tocframe template=<your-name>}`.

`edging` Apply (with `edging`) or deactivated (`noedging` when `\gothamset{edging default=on noedging }` was given) the sidebar canvas left (and right respectively) template, through `\defbeamertemplate{sidebar canvas left}{default/<your-name>}{<your-def>}` (and right) and the option `\gothamset{sidebar canvas left template=<your-name>}` (and right).

`nologo` Apply empty logo (and `footline`, `footer` respectively) templates. This can be convenient when extra space is needed.

`nofooter`

`rotateFooter` Enable or disable the rotation of the elements in the footer. This can be convenient when extra space is needed in 16/9 mode especially.

`noRotateFooter`

`part` Apply `\usebeamertemplate{part frame}`, (`section frame`, `subsection frame` and `section subsection frame` respectively) templates. This makes more sense for internal use, but can be reused everywhere else.

`subsec`

`subsubsec`

`tocpart` Apply `\usebeamertemplate{toc part frame}`, (`toc section frame`, `toc subsection frame` and `toc subsection frame` respectively) templates. This makes more for internal use, but can be reused everywhere else.

`tocsubsec`

`tocsubsubsec`

3.4 Commands for Customization

Gotham defines some commands that can be used as hooks, ie., that can be redefined when needed.

\gothamInstituteLogoCircle[1] [4ex]

#1 : height of the picture.

Command to include the circular logo of your institute in the frametitle when using `\gothamset{progressbar position=circlehead}`. For example, you redefine this command through:

```
\renewcommand{\gothamInstituteLogoCircle}[1][4ex]{
  \includegraphics[height=#1]{<your-logo-circular>}
}
```

\gothamInstituteLogoSquare[1] [4ex]

#1 : height of the picture.

Command to include the circular logo of your institute in the frametitle. For example, you redefine this command through:

```
\renewcommand{\gothamInstituteLogoSquare}[1][4ex]{
  \includegraphics[height=#1]{<your-logo-square>}
}
```

\gothamFrameSubtitleSep Command defining the separator used between the frametitle and the subtitle.

\gothamRightFiligrane \gothamLeftFiligrane Commands that are used by default in the `sidebar canvas right` (and left). This avoids the redefinition of the whole templates, especially since the `sidebar canvas right` is containing elements by default in Beamer theme (like the logo).

\gothamtitlepagelogo The command to insert the institute logo(s) on title page. For example, you redefine this command through:

```
\renewcommand{\gothamtitlepagelogo}{
  \includegraphics[height=#1]{<your-logo-on-titlepage>}
}
```

`\gothamtitlepagebg` The command to insert the background title page. For example, you redefine this command through:

```
\renewcommand{\gothamtitlepagebg}{
  \includegraphics [height=#1]{<your-background-on-titlepage>}
}
```

`\partContentName` The command to change the title of frames containing part's (or section or subsection) table of contents. For example, you redefine this command through:

`\secContentName`
`\subsecContentName`

```
\renewcommand{\partContentName}{Part 's agenda}
```

`\partpageOptions` The commands to change the options of frames containing part's title (or section or subsection) at its start. For example, you redefine this command through:

`\sectionpageOptions`
`\subsectionpageOptions`
`\subsubsectionpageOptions`

```
\renewcommand{\partpageOptions}{plain}
```

New: 2025-01-06

`\partpageTocOptions` The command to change the options of frames containing part's (or section or subsection) table of contents. For example, you redefine this command through:

`\sectionpageTocOptions`
`\subsectionpageTocOptions`

```
\renewcommand{\partpageTocOptions}{nofooter}
```

New: 2025-01-06

`\partTocOptions` The command to change the options for part's (or section or subsection) table of contents at the beginning of this part. For example, you redefine this command through:

`\sectionTocOptions`
`\subsectionTocOptions`

```
\renewcommand{\sectionTocOptions}{
  currentsection, hideallsubsections
}
```

New: 2025-01-06

gothamZerosectionframes Boolean variable to flag if they are frame in a zeroth section. This variable helps to adapt the spreading of `mini frames` nav bar. This variable is automatically set if the spread is set correctly at the beginning of the presentation. If the spreading or the mini frame nav is disable at the zeroth section then reactivated latter, it might create unwanted spreading. In such situation the variable has to be set manually to correct the spreading. You change its value with `\booltrue{gothamZerosectionframes}`.

darkBG
transparentBG A boolean variables that are true with the background mode. It can be useful when you want to apply different codes according to the background you are on. You can use it as in the following example:

```
\ifbool{darkBG}{code to apply on dark background}{other code
for light or transparent backgrounds}
```

\gothamHookFooter The command to add elements (like a logo) in the footer without redefining it completely. For example, you redefine this command through:

Updated: 2024-11-11

```
\renewcommand{\gothamHookFooter}{
  \usebeamertemplate{instituteLogo}
}
```

\gothamHookPostColorBGSet The command to change the colors that are related to the background color setting (ie. frametitle, standin, standout and titlepage). For example, you redefine this command through:

Updated: 2024-11-11

```
\renewcommand{\gothamHookPostColorBGSet}{
  \colorlet{colorFrametitle}{yourColor}
  \colorlet{colorStandout}{yourColor}
  \colorlet{colorStandin}{yourColor}
  \colorlet{colorTitlepage}{yourColor}
  \setbeamercolor{frametitle}{fg=yourRed, bg=}
}
```

3.5 Color Customization

The included **Gotham** color theme is used by default, but its colors can be easily changed to suit your tastes. All of the theme's styles are defined in terms of three Beamer colors:

- `normal text` (dark fg, light bg)
- `alerted text` (colored fg, should be visible against dark or light)
- `example text` (colored fg, should be visible against dark or light)

An easy way to customize the theme is to redefine these colors using

```
\setbeamercolor{ ... }{ fg= ... , bg= ... }
```

in your preamble. For greater customization, you can redefine any of the other stock Beamer colors. In addition to the stock colors the theme defines a number of **Gotham** specific colors, which can also be redefined to your liking.

```
\setbeamercolor{progress bar}{ ... }  
\setbeamercolor{title separator}{ ... }  
\setbeamercolor{progress bar in head/footer}{ ... }  
\setbeamercolor{progress bar in section page}{ ... }
```

3.6 Font Customization

The default font for **Gotham** is Fira. This can be easily changed using the standard font selection commands of the `fontspec` package. So if you prefer, for example, the **Ubuntu** font family, just add the following two commands after loading the **Gotham** theme.

```
\setsansfont{Ubuntu}  
\setmonofont{Ubuntu Mono}
```

If you are expecting to present in a large room or with an underpowered projector, you may want to change the font to a heavier weight of Fira to maximize readability.

```
\setsansfont[BoldFont={Fira Sans SemiBold}]{Fira Sans  
Book}
```

3.6.1 Old style figures

The regular `fontspec` mechanism for changing glyph appearance also applies to this theme. If you want to have old style figures in the text but regular lined figures for math, you could add the following to your preamble:

```

\usefonttheme{professionalfonts} % required for
  mathspec
\usepackage{mathspec}
\setsansfont[BoldFont={Fira Sans},
  Numbers={OldStyle}]{Fira Sans Light}
\setmathsfon(Digits)[Numbers={Lining, Proportional}]{
  Fira Sans Light}

```

3.7 Length Customization

```

\gothamFrametitleToppading
\gothamFrametitleBottompadding
\gothamFrametitleLeftpadding
\gothamFrametitleRightpadding

```

Lengths controlling the padding around the frametitle.

```

\gothamFramesubtitleStrutend

```

Lengths controlling the padding around the frametitle.

```

\sidebarRightHOffset
\sidebarLeftHOffset

```

Length controlling the horizontal and vertical offset in order to position `\gothamRightFiligrane` (respectively `\gothamLeftFiligrane`) when using the default sidebar canvas (right and left) from **Gotham**.

```

\gothamFootlineRuleLeftPadding
\gothamFootlineRuleHeight
\gothamFootlineRuleLength

```

`\gothamFootlineRuleLeftPadding` is controlling the horizontal space between the left border of the page and the left side of the rule. `\gothamFootlineRuleHeight` controls the height and `\gothamFootlineRuleLength` the length of the rule used to delimit the footer.

`\gothamFootlineHRightOffset`
`\gothamFootlineVOffset`
`\gothamFootlineHeight`
`\gothamFootlineDepth`

Updated: 2024-11-11

`\gothamFootlineHRightOffset` is controlling the horizontal space between the right border of the page and the side of the footline. `\gothamFootlineVOffset` is controlling space between to bottom of the text (or the footnote) and the footline. `\gothamFootlineHeight` and `\gothamFootlineDepth` are controlling the height and depth of the footline baseline.

`\gothamLeftFooterPadding` `\gothamLeftFooterPadding` is controlling the horizontal space between the left border of the page and the side of the footer. `\gothamRightFooterPadding` is controlling the horizontal space between the right the footer and the page number. `\gothamFooterHOffset` is controlling the horizontal space between the footer and the bottom of the page (or the progressbar).

`\gothamHposLeftRotFooter` Lengths that are controlling the horizontal and vertical positioning of the left and right
`\gothamHposRightRotFooter` elements of the rotated footer.
`\gothamVposLeftRotFooter`
`\gothamVposRightRotFooter`

`\gothamProgressCircHeight`
`\gothamCounterCircleRadius`
`\gothamProgressCircBorderWidth`

Lengths controlling the aspect of `progress circle`. `\gothamProgressCircHeight` is controlling the inner height of the circle (related to its diameter). `\gothamCounterCircleRadius` is controlling the size of the counter circle containing the frame number. `\gothamProgressCircBorderWidth` is controlling width of the progress circle.

`\gothamProgressHeadFootLineheight`

`\gothamProgressHeadFootLineheight` is controlling the height of the progress bar in normal frames (or its width when it is put in side bars).

`\gothamCircleNumberingVshift`
`\gothamCircleNumberingHshift`

Lengths controlling the vertical and horizontal positioning of the `circle` frame numbering template.

`\sectionoffset`
`\gothamProgressSectionHeight`

`\sectionoffset` controls the horizontal offset of the (section title + progress bar) block. Can be useful when extra stuff wants to be displayed on sides of the block. The default value is `0pt`.

`\gothamProgressSectionHeight` controls the height of the progress bar used in part/-section/subsection/subsubsection frames (when progress bar are used).

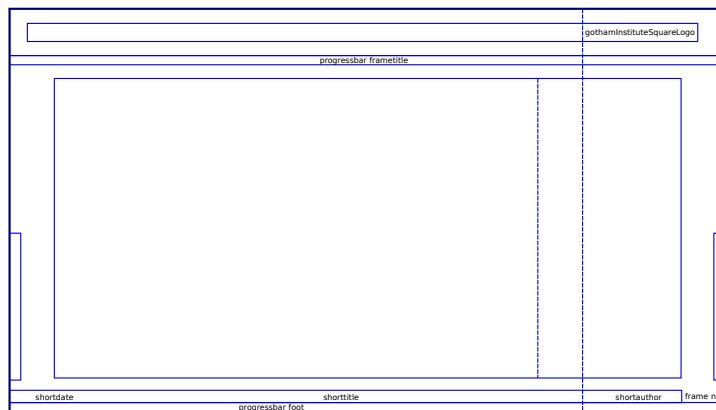


Figure 3: The layout used by **Gotham**.

4 Tips & Tricks

4.1 Backup Slides

Speakers will often include extra slides at the end of their presentation to refer to during audience questions. One easy way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

Gotham will automatically turn off slide numbering for slides in the appendix.

4.2 Sources of inspiration

Many Beamer themes have been used as sources of inspiration to build **Gotham**:

- <https://github.com/matze/mtheme> for dark/light theme and standout page.

- <https://github.com/hamaluik/Beamer-Theme-Execushares> and <https://github.com/pcafrica/focus-beamertheme> for titlepage and section-page.
- <https://github.com/LukasPietzschmann/awesome-beamer> for titlepage.
- <https://github.com/luistar/unina-beamer/tree/master> and <https://github.com/jkjaer/aauLatexTemplates/tree/master/aauBeamer/aausimple> for circlehead progress bar.
- <https://github.com/povanberg/flux-beamer> for toc.
- <https://github.com/fliptanedo/FlipBeamerTheme> for numbering circled fraction.
- https://gitlab.com/RomainNOEL/latex3_template_pkg for LaTeX3 template.

5 Known Issues

5.1 Title formats

Be aware that not every font supports small caps, so the `smallcaps` or `upper` options may not work if you use a font other than **Fira Sans**. In particular, the Computer Modern sans-serif typeface, which is used when **Gotham** is compiled with pdfL^AT_EX, does not have a small-caps variant.

The title format options `upper` and `smallcaps` are quite nice from an aesthetic point of view, but their use of `\MakeLowercase` and `\MakeUppercase` can cause unexpected problems. For example:

- Some commands, like `\`, do not work inside `\MakeLowercase` and `\MakeUppercase`. (See [#125](#))
- Only alphabetic characters are affected by `\MakeLowercase`, so numerals and punctuation remain at full height. This can spoil some of the aesthetic benefits of `upper`. (See [#33](#))
- `\MakeLowercase` and `\MakeUppercase` apply to math mode and `\scshape` does not. This can easily introduce mathematical errors that are hard to catch.
- It is impossible to typeset symbols which are encoded as uppercase letters in a different font. In particular, `\mathbb` and `\mathcal` letters will be replaced by other math glyphs. (See [#153](#))

The `smallcaps` and `upper` options are safe to use if your titles contain only alphabetic characters and do not require the expansion of any macros.

5.2 Interactions with other color themes

Gotham can be used along with any other Beamer color theme, such as **crane** or **seahorse**. If you wish to do this, it is usually best to include the **Gotham** subpackages individually so the **Gotham** color theme is never loaded. This will prevent conflicts between the **Gotham** color theme and your preferred theme.

For example, overriding the color theme as follows may not work as expected because `\usetheme{gotham}` loads the **Gotham** color theme, which defines a relationship between the frametitle background and the primary palette of the theme. Since **seahorse** assumes a different relationship between its palettes, the result is a grey, rather than periwinkle, frametitle background.

```
\usetheme{gotham}
\usecolortheme{seahorse}
```

The correct colors are chosen if the **Gotham** outer, inner, and font themes are loaded separately:

```
\useoutertheme{gotham}
\useinnertheme{gotham}
\usefonttheme{gotham}
\usecolortheme{seahorse} % or your preferred color
                           theme
```

Please note that **Gotham** may not use all the colors defined in your favourite Beamer color theme. In particular, **Gotham** does not set a background color for the title; this will cause issues when using color themes like **whale** which set a white foreground for the title.

5.3 Notes on second screen

If you use the `[show notes on second screen]` option built into Beamer and compile with XeLaTeX, text on slides following the first section slide may be rendered in white instead of the regular color. This is due to a **bug** in Beamer or XeLaTeX itself. You can work around it either by compiling with LuaTeX or by adding the following code to your preamble to reset the text color on each slide.

```
\makeatletter
\def\beamer@framenotesbegin{% at beginning of slide
  \usebeamercolor[fg]{normal text}
  \gdef\beamer@noteitems{}}%
```



```

\gdef\beamer@notes{}%
}
\makeatother

```

5.4 Standout frames with labels

Because the `standout` frame option creates a group to restrict the colour change to a single slide, labels defined after calling `standout` will stay local to the group. In other words, the following may result in a “label undefined” error.

```

\begin{frame}[standout, label=conclusion]{Conclusion}
  Awesome slide
\end{frame}

```

To fix this problem, change the order of the keys in the frame.

```

\begin{frame}[label=conclusion, standout]{Conclusion}
  Awesome slide
\end{frame}

```

This error can be unwittingly triggered if you export your slides from Emacs Org mode, which automatically adds labels after frame options. Alex Branham [offers](#) the following solution for Org mode users, using `org-set-property`.

```

* Start of a frame
:PROPERTIES:
:BEAMER_opt: label=conclusion,standout
:END:

```

5.5 Standout frames with Pandoc

With Pandoc versions prior 1.17.2 it was not possible to create standout frames because Pandoc only supported a specific list of frame attributes thus ignoring additional attributes such as `{.standout}`.

5.6 Other issues

- `enumitem` is not working with Beamer.

Todo

List of thing that could be improved (any volunteer welcome):

- Reduce the number of @ .
- Rename a lot of length with gotham at the beginning of their name.
- Turn internal length into `_dim`.
- Improve documentation.
- Add a hexagonal, wavy and add lengths on the blueprint layout backgrounds.
- Remove colors from tests inner and outer.
- Replace the `\setbeamertemplate{yy}[default/xx]` by `__gotham_inner_set_template:nn{tit` or merge them because the default values in dict/template are interesting but `__gotham_inner_set_template` are simpler.
- add colorset more **blue-ish**, **green-ish** ... from colorA etc.
- add Gotham to lists of Awesome Beamer themes.

6 License

Gotham is licensed under the terms of the [LaTeX project public license \(LPPL\) 1.3c](#) license.