
Ma première présentation avec Beamer

Germain Vallverdu

Septembre 2008

germain_vallverdu@yahoo.fr

Le but de ce petit texte est de regrouper au même endroit ce que j'ai cherché à droite à gauche sur internet pour faire ma première présentation avec Beamer. Sur internet il existe des choses beaucoup plus complètes, notamment le manuel (250 pages) [1], et beaucoup plus détaillé. Ce document contient les commandes que j'ai utilisées et quelques exemples qui j'espère, aideront à faire une première présentation avec beamer.

Il est possible de compiler le fichier TeX avec `latex` ou `pdflatex` qui présente le double avantage de supporter les formats images bitmaps, jpeg, png, pdf et de créer directement un fichier pdf.

Table des matières

1	Le préambule	1
1.1	Déclaration de la classe beamer	1
1.2	Choix du thème	1
1.3	Page de titre	2
1.4	Définition d'un logo	2
1.5	Divers	2
2	Créer la présentation	2
2.1	Créer une diapositive	2
2.2	Remplir les diapositives	3

2.2.1	liste	3
2.2.2	Les blocks	3
2.3	Overlay - Apparition progressive du contenu d'une frame	4
2.3.1	Les différentes commandes	4
2.3.2	Cas particulier des listes	4
3	Boite à outils	5
3.1	Package xcolor [4]	5
3.2	Package animate [5]	5
3.3	Package textpos [6]	5
3.4	Package hyperref	6
3.5	Le package picture	6
3.6	La barre de navigation	6
3.7	Options pdf	6

1 Le préambule

1.1 Déclaration de la classe beamer

```
\documentclass[9pt]{beamer}
```

A savoir que :

- La classe beamer crée une page en format paysage de 128mm de large sur 96mm de haut.
- La taille de police agit sur l'ensemble des éléments : titre de la présentation, titre des diapositives etc ...
- On peut rajouter des options comme `draft` pour accélérer la compilation.

1.2 Choix du thème

Le thème définit le jeux de couleurs et la présentation de la page qui sera utilisé pour la présentation. On peut soit utiliser une thème global qui définira l'allure de l'ensemble de la présentation, soit préciser l'utilisation d'un thème différent pour chaque élément :

Thème	Commentaire	utilisation
global	Pour l'ensemble des éléments de la présentation	<code>\usepackage{beamerthemeWarsaw}</code>
Les couleurs	sans commentaire	<code>\usecolortheme{seagull}</code>
les polices	sans commentaire	<code>\usefonttheme{serif}</code>
innertheme	Thème pour l'intérieur de la diapositive : liste, block, tableau etc ...	<code>\useinnertheme{circle}</code>
outertheme	Thème pour l'extérieur de la diapositive : Titre de la diapositive, entête, pied de page etc ...	<code>\useoutertheme{shadow}</code>

L'ensemble des thèmes disponible se trouve dans le dossier où beamer est installé. Par exemple dans le dossier `/usr/share/texmf/tex/latex/beamer/themes/`.

1.3 Page de titre

La page est créée par la commande `\titlepage`.

Dans le préambule on peut définir les éléments suivant :

```
\title[titre court]{titre long}
\author[court]{long}
\institute{ Université ...}      % votre labo, entreprise etc ...
\date{ 1er mai }                 % une date ou \today pour la date du jour
```

Le titre long est sur la première diapositive et le titre court est dans les pieds de page sur toutes les autres.

Pour les auteurs c'est le même principe que le titre long et le titre court. Cela permet par exemple de différencier le nom de celui qui présente (écrit sur toutes les diapositives) et des gens qui ont participé au travail (écrit sur la première diapositive).

1.4 Définition d'un logo

Il est possible de définir une image en tant que logo qui sera pré sen sur toutes les diapositives à un endroit qui va dépendre du thème choisi.

```
\logo{\includegraphics[height=5mm]{img/logo.png}}
```

1.5 Divers

Les packages usuels de latex fonctionnent avec beamer. Voici par exemple quelques packages classiques :

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}      % pour les accents (mettre latin1 pour
                                  % windows au lieu de utf8)
\usepackage[frenchb]{babel}     % le documents est en français
\usepackage{amsmath}            % un packages mathématiques
\usepackage{xcolor}             % pour définir plus de couleurs
\usepackage{graphicx}           % pour insérer des figures
```

2 Créer la présentation

2.1 Créer une diapositive

Chaque diapositive est simplement définie par l'environnement `frame`. Par exemple les deux premières diapositives, sont construites par le code :

```
% page de titre
\begin{frame}

    \titlepage

\end{frame}

% deuxième diapositive
\begin{frame}

    \frametitle{La deuxième diapositive}

    \begin{center} bla bla bla .... \end{center}

\end{frame}
...
```

La commande `\frametitle{ ... }` permet de définir le titre de la diapositive. Par défaut les diapositives sont centrées verticalement. Ainsi "bla bla bla ..." apparaîtra au centre de la diapositive.

2.2 Remplir les diapositives

Les diapositives peuvent contenir tout ce que vous avez l'habitude de mettre dans une document latex classique : tableau, listes, image.

2.2.1 liste

Les listes `itemize` ont des items spécifiques à beamer dont la couleur peut varier en fonction du thème choisi : défaut (triangle), triangle, square, ball, circle. On peut choisir les items ainsi que les sous items en les définissant dans le préambule de la façon suivante :

```
\setbeamertemplate{itemize item}[ball]
\setbeamertemplate{itemize subitem}[triangle]
\setbeamertemplate{itemize subsubitem}[circle]
...
```

Voici un exemple des différentes puces disponibles :

type	rendu
triangle	▶ un triangle
ball	● une balle
circle	● un cercle
square	■ un carré

Un exemple des autres listes :

Code	Rendu
<pre>\begin{enumerate} \item item 1 \item item 2 \item item 3 \item item 4 \end{enumerate}</pre>	<ol style="list-style-type: none">1 item 12 item 23 item 34 item 4
<pre>\begin{description} \item[item1] description 1 \item[item2] description 2 \item[item3] description 3 \item[item4] description 4 \end{description}</pre>	<p>item1 description 1 item2 description 2 item3 description 3 item4 description 4</p>

2.2.2 Les blocks

Beamer propose un environnement `block` qui permet de mettre des éléments en valeur. Il existe trois environnements blocks différents, les blocks simples `block`, alerte `alertblock` et exemple `exampleblock`. Ils s'utilisent simplement de la façon suivante :

```
\begin{block}{ titre du block }
  Texte, équations, image, tableau etc ...
\end{block}
```

```
\begin{alertblock}{ titre du block }
  Texte, équations, image, tableau etc ...
\end{block}
```

Pour avoir des blocks arrondis avec une ombre, rajouter la commande suivante dans le préambule :

```
\setbeamertemplate{blocks}[rounded][shadow=true]
```

block normal

texte, image tableau ect ...



block alerte

ATTENTION!!

block exemple

Un exemple !

On peut modifier les couleurs des blocks, en indiquant le code suivant soit dans le préambule, soit localement avant d'utiliser un block :

```
\setbeamercolor{block title}{fg=black,bg=blue1} %titre block normal
\setbeamercolor{block body}{fg=black,bg=bleu1!50} %corps block normal
% idem pour un block alerte
\setbeamercolor{block body alerted}{fg=white,bg=red}
```

Le !70 permet d'ajuster la couleur, de foncé (100) à plus clair (0). Il n'est pas obligatoire de donner un titre, ce qui permet d'encadrer un mot ou une phrase. On peut contrôler la largeur des block en les insérant dans une boîte (`textblock` ou `minipage`). Voici un exemple avec le block alerte :

```
\setbeamercolor{block body alerted}{fg=white,bg=red}
\begin{minipage}{50mm}
  \begin{alertblock}{}
    \begin{center}
      \textbf{Très important}
    \end{center}
  \end{alertblock}
\end{minipage}
```

Très important

L'utilisation de `textblock` permet en plus de positionner le block où on le veut sur la page.

2.3 Overlay - Apparition progressive du contenu d'une frame

Lors d'une présentation il est souvent préférable de faire afficher les éléments au fur et à mesure à l'aide des overlay. Il faut alors faire une différence entre une **frame** et une diapositive. Les overlays permettent d'afficher en plusieurs fois le contenu d'une **frame** (délimité par l'environnement `frame`) en créant plusieurs diapositives. Pour cela, la classe `beamer` met à disposition plusieurs commandes : `\uncover<>`, `\(in)visible<>`, `\only<>`. Entre les `< >`, il faut indiquer l'ordre d'apparition des éléments. Beamer construira ensuite le nombre de diapositives nécessaire en faisant apparaître progressivement les éléments.

<code><1-></code>	agit sur la 1ère diapositive de la frame et jusqu'à la dernière.
<code><2-></code>	à partir de la deuxième diapositive et jusqu'à la fin.
<code><-2></code>	jusqu'à la deuxième diapositive.
<code><2></code>	uniquement sur la deuxième diapositive.
<code><3-5></code>	de la troisième à la cinquième diapositive.
...	

Le décompte des diapositives est remis à zéro pour chaque nouvelle **frame**. Pour que le texte qui est présent sur une frame mais qui ne s'affiche pas sur les premières diapositives s'affiche en beaucoup plus clair et peu lisible il faut rajouter la commande suivante dans le préambule :

```
\beamertemplatetransparentcovered
```

2.3.1 Les différentes commandes

<code>uncover<n->{texte}</code>	affiche les éléments entre <code>{}</code> à partir de la diapositive <code>n</code>
<code>only<n->{texte}</code>	affiche les éléments entre <code>{}</code> à partir de la diapositive <code>n</code> mais la place prise par le texte n'était pas réservée sur les diapositives précédentes. La mise en page est réorganisée pour faire apparaître le texte.
<code>(in)visible<-n>{texte}</code>	idem que <code>uncover</code> (ou inverse).
<code>alert<4->{texte}</code>	le texte entre <code>{}</code> sera en rouge à partir de la quatrième diapositives.

Pour certains éléments (comme des figures ou des `textblocks`) `uncover` ne fonctionne pas et il faut utiliser `(in)visible` à la place. D'autre part si on a choisi que le texte s'affiche en plus clair et peu lisible avant qu'il soit réellement présent, il faut utiliser `uncover`. En effet, l'utilisation de `(in)visible` le fait complètement disparaître.

La commande `alt<n>{texte1}{texte2}` offre la possibilité de faire une alternative en fonction de la diapositive. Elle permet de substituer `texte1` et `texte2` suivant la diapositive, exemple :

```
\alt<3>{Je suis sur la diapositive 3}{Je ne suis pas sur la diapositive 3}
```

Toutes ces commandes fonctionnent très bien pour du texte, mais elles peuvent également agir sur des figures, des lignes ou colonnes d'un tableau etc ... Lorsqu'on utilise des boîtes (`textblock`, `parbox`, `minipage` etc ...) il est préférable de les mettre à l'intérieur de la boîte pour qu'elles agissent directement sur le contenu et pas sur la boîte.

2.3.2 Cas particulier des listes

Pour les listes, on peut utiliser les commandes précédentes mais d'autres sont disponibles. Les deux codes suivants produisent la même chose. Beamer fabrique 4 diapositives en faisant successivement apparaître les 4 items.

<code>\begin{enumerate}[<+>]</code>	<code>\begin{enumerate}</code>
<code>\item item 1</code>	<code>\item<1-> item 1</code>
<code>\item item 2</code>	<code>\item<2-> item 2</code>
<code>\item item 3</code>	<code>\item<3-> item 3</code>
<code>\item item 4</code>	<code>\item<4-> item 4</code>
<code>\end{enumerate}</code>	<code>\end{enumerate}</code>

La première version (gauche) code pour un affichage séquentiel de manière automatique. La seconde offre plus de souplesse dans l'ordre d'apparition et de disparition des éléments.

3 Boite à outils

3.1 Package xcolor [4]

Ce package donne accès à plus de couleurs disponibles et permet d'en définir de nouvelles en utilisant les codes RVB, CYMK, ou HTML avec la syntaxe suivantes :

```
\definecolor{monred}{HTML}{9D0909}
\definecolor{monbleu}{RVB}{0,0,1}
```

3.2 Package animate [5]

Ce package permet d'insérer des animations dans un document pdf. D'une manière simple, l'animation est créée comme une succession d'image, comme un dessin animé. Cependant, le package permet aussi d'insérer des vidéos mais celle-ci ne sont pas directement encapsuler dans le pdf et doivent donc être accessible. Voici un exemple de l'utilisation de `animate` :

```
\animategraphics[autoplay,loop,height=10mm]{10}{tux_ask}{1}{2}
```

L'animation est créée à l'aide de deux images `tux_ask1.jpg` et `tux_ask2.jpg`. L'option `autoplay` fait démarrer l'animation toute seule, et `loop` permet de jouer l'animation en boucle. Le chiffre 10 est la vitesse d'enchaînement des images (normalement en nombre d'images par seconde). On donne ensuite la racine du nom des images (ici `tux_ask`) et le numéro des images, ici de 1 à 2. L'animation est encapsulée dans le pdf qui est donc auto-suffisant.

Remarque : Une petite contrainte, il est nécessaire d'utiliser acrobat reader pour que ça fonctionne.

3.3 Package textpos [6]

Ce package est très utile pour préparer sa présentation car il permet, à l'aide de coordonnées (x,y) , de positionner des objets où on le souhaite sur la page. Il s'utilise de manière simple avec un environnement `textblock*` qui s'utilise un peu comme une `minipage` ou une `parbox`. Pour l'utilisation il faut mettre ça dans le préambule :

```
\usepackage[absolute,showboxes,overlay]{textpos}
\textblockorigin{x}{y} % origine des positions
\TPshowboxestrue % affiche le contour des textblock
\TPshowboxesfalse % n'affiche pas le contour des textblock
```

L'origine des coordonnées est donnée par rapport au coin supérieur gauche de la page. L'option `absolute` permet de placer tous les blocks de la page par rapport à cette origine. Ainsi, si on déplace quelque chose dans la page les autres blocks ne sont pas déplacés. L'option `overlay` permet aux différents blocks ou autres éléments de la page de se superposer. Cette option est vraiment nécessaire, si on ne la met pas et que deux blocks se superposent, l'un d'eux sera invisible. L'option `showboxes` en relation avec les commandes `TPshowboxes(false)true` permet de faire afficher ou non une bordure aux blocks. Cela peut être utile pendant la conception des diapositives pour mieux se rendre compte de où se place les blocks.

Il suffit ensuite d'utiliser l'environnement `textblock*` dont la syntaxe générale est :

```
\begin{textblock*}{largeur}[h,v](x,y)
    tout ce qu'on veut !!
\end{textblock*}
```

Les paramètres sont assez explicite, `largeur` est la largeur du block. `x,y` sont les coordonnées du block par rapport à l'origine, elles sont données entre parenthèses. Entre crochets, `h` et `v` définissent le point du block qui aura les coordonnées `x` et `y`, il n'est pas obligatoire de les préciser. Par défaut `h` et `v` valent zéro et définissent le point en haut à gauche du block. Ces grandeurs sont relatives, par exemple le centre du block est défini par `[0.5,0.5]` le coin en bas à gauche par `[0,1]` ainsi de suite.

L'avantage d'utiliser l'environnement `textblock*`, avec une `*`, est que la largeur et les coordonnées `x` et `y` sont directement données dans l'unité désirée. Si on ne met pas l'étoile il faut définir des unités dans le préambule avec les commandes suivantes [6] :

```
\setlength{\TPHorizModule}{1mm}
\setlength{\TPVertModule}{\TPHorizModule}
```

La largeur et les coordonnées `x` et `y` sont alors données comme multiples de `TPHorizmodule` et `TPVertModule`.

Remarques :

Les `textblocks` ne prennent pas de place sur la page. Si on place un block en haut à gauche et qu'avant ou après on écrit du texte, le block et le texte seront superposés. Pour placer le texte en dessous du block il faut soit le placer lui aussi dans un `textblock`, soit utiliser la commande `\rule{largeur}{hauteur}` ou `\vskip` pour décaler le texte verticalement.

Si on place plusieurs `textblock` sur la page, qu'ils se superposent et que leurs contenu n'est pas transparent, c'est le premier block qui sera par dessus les autres et ainsi de suite.

3.4 Package hyperref

Le package `hyperref` s'utilise comme dans un document \LaTeX classique et offre les mêmes fonctionnalités.

3.5 Le package picture

Ce package s'utilise comme dans un document \LaTeX classique et permet notamment de faire des flèches et des lignes. On peut ensuite positionner ces éléments sur la page en insérant l'environnement `picture` dans un environnement `textblock`. Voici la syntaxe générale de l'environnement `picture` :

```
\begin{picture}(largeur,hauteur)
  \thicklines
  \put(0,0){\line(1,0){10}}
  \put(10,0){\vector(0,1){10}}
\end{picture}
```



La largeur et la hauteur, ainsi que toutes les dimensions ou coordonnées à l'intérieur sont des multiples d'une longueur définies dans le préambule par :

```
\setlength{\unitlength}{1mm}
```

La commande `put(x,y){objet}` place l'objet au point de coordonnées x et y , l'origine étant en bas à gauche de la boîte `picture`. L'objet `\line(x,y){longueur}` crée une ligne de longueur `\longueur` et dont la direction est donnée par le vecteur de coordonnées (x,y) . Par exemple $(0,1)$ définit une ligne verticale et $(1,1)$ une ligne oblique à 45° . L'objet `\vector` est identique à `\line` dans son utilisation mais il crée une flèche à la place. La commande `\color{couleur}` permet de choisir la couleur des objets.

3.6 La barre de navigation

Vous avez sans doute remarqué que les présentations Beamer affiche (souvent en bas à droite) une série de petits icônes permettant de se déplacer d'une diapositive à une autre. On peut supprimer l'un ou l'autre de ces icônes en commentant les lignes voulues dans le code suivant à placer dans le préambule :

```
\setbeamertemplate{navigation symbols}{
  \insertslidenavigationsymbol
  \insertframenavigationsymbol
  \insertsubsectionnavigationsymbol
  \insertsectionnavigationsymbol
  \insertdocnavigationsymbol
  \insertbackfindforwardnavigationsymbol
}
```

3.7 Options pdf

Voici une série de commandes, à mettre dans le préambule, qui permettent de définir différents paramètres du fichier pdf.

```
\hypersetup{
  pdfpagemode = FullScreen,% afficher le pdf en plein écran
  pdfauthor   = {auteur(s)},%
  pdftitle    = {Titre ...},%
  pdfsubject  = {Sujet ...},%
  pdfkeywords = {Mots clés séparés par des virgules},%
  pdfcreator  = {PDFLaTeX},%
  pdfproducer = {PDFLaTeX}%
}
```

Références

- [1] [Beamer homepage](#), le manuel et le package beamer.
- [2] La [FAQ \$\text{\LaTeX}\$](#) grappa de l'université de Lille.
- [3] Un site de l'ens avec une aide \LaTeX et beamer, [tuteur ens](#).
- [4] Le package `xcolor` avec le [manuel](#).
- [5] Le package `animate` et le [manuel](#).
- [6] Le package `textpos` et le [manuel](#).
- [7] Le site de Mc Collilieux, très bien fait [introbeamer](#).