

LaTeX et beamer

jml

12 novembre 2004

Table des matières

1	Introduction	1
2	Installation	1
3	Mise en oeuvre	1
4	Code source entête	2
4.1	Classe de document	2

1 Introduction

Introduction

Beamer utilise divers environnements qui permettent la création de:

- Fichier de présentation au format .pdf à exploiter au vidéo projecteur <presentation>
- L'équivalent en forme condensé pour impression d'un document papier <handout>
- L'équivalent de la forme condensée mais avec les notes destinées à l'orateur <handout + notes>
- Et enfin un fichier .pdf format article reprenant le cours de l'exposé <article>

Le tout en partant d'un seul et unique fichier source ! Ne cherchez donc pas plus loin la raison de mon intérêt pour la classe beamer !

2 Installation de beamer

Installation

Le fonctionnement de beamer est conditionné par l'installation de:

pgf, le package de gestion de graphique

xcolor, le package de gestion des couleurs

Le package pgf est un équivalent de graphicx spécifique à beamer.

Installation automatique

Les utilisateurs de DEBIAN et MANDRAKE disposent d'un package beamer. Ceux de FreeBSD peuvent utiliser le port /print/latex-beamer. Le lecteur se reportera utilement à la documentation de sa distribution pour les instructions d'utilisation des installeurs. Pour ma part je n'ai pas utilisé l'installation automatisée, je ne peux donc donner aucun avis sur l'efficacité de ces méthodes

Installation manuelle

Pour les distributions qui ne disposent pas d'un package il reste la possibilité de réaliser une installation manuelle. Procéder comme suit:

- Récupérer latex-beamer-2.21.tar.gz (ou plus récent)
- Idem pour pgf-0.62.tar.gz et xcolor-2.00.tar.gz
- Décompacter dans un sous rép. temporaire
- Copier les nouveaux sous rép créés, beamer, pgf et xcolor dans /usr/local/texmf/tex/latex/

- Sous compte root lancer texhash pour mettre à jour la base de données LaTeX

3 Mise en oeuvre

Préambule

La structure d'un fichier beamer est double:

- Une structure classique section + sous section
- Une suite de "frame", chaque frame étant une base de calque

Sur chacune des bases de calques vont venir s'empiler les couches qui vont générer l'animation.

La structure classique section + sous section semble limitée à deux niveaux. Comme dans LaTeX les titres de section vont constituer une table des matières.

4 Le code source de l'entête de document

4.1 Choix de la classe de document

Classe de document

La syntaxe de la définition de classe de document prendra l'une des formes suivantes:

```
\documentclass[10pt, green]{beamer}
% version standard
%\documentclass[class=article, a4paper]{beamer}
% article destiné à l'impression
%\documentclass[10pt, green, handout]{beamer}
% version imprimable pour assistance
%\documentclass[10pt, green, handout, ->
notes=show]{beamer}
% version imprimable avec notes d'orateur
```

-> indique un renvoi de mise en page

On décommentera la ligne correspondant à la forme de document souhaitée.

La syntaxe est relativement simple. Quelques commentaires:

10 pts fixe la taille de la police de caractères, ici 10 pts. Ne pas perdre de vue que beamer génère des "images" de 12*9 cm environ. La taille de la police doit être appréciée en correspondance.

green détermine la couleur de fond du document, ici toutes les adaptations sont permises.

class=article est le mot clef qui conduit pdflatex à créer un document selon la charte article classique LaTeX.

handout désigne une compilation qui produit une version imprimable du document projeté. Très schématiquement tous les calques qui se superposent sur un fond donné sont gelés en position pour donner une image unique.

notes=show va générer des pages intermédiaires entre les frames, pages contenant les notes de l'orateur. Cette option ne présente, à mon avis, d'intérêt que combinée avec handout.

Pour passer d'une version à la suivante il suffit de commenter / décommenter judicieusement les lignes pour en laisser une seule active. La compilation se passe alors sans problème, sauf la compilation de l'article qui charogne un peu. Deux parades, soit forcer le passage en validant puis refaire une seconde compilation qui passera sans problème (si le code est correct), soit supprimer au préalable le fichier .toc.