



Lighter : une alternative légère et descriptive à Beamer

Contexte

Beamer est l'outil de génération de *slides* de présentation le plus couramment utilisé dans l'écosystème **LaTeX**. Cet outil a des avantages par rapport aux outils **WYSIWYG** de création de *slides* (**LibreOffice Impress**, **Microsoft PowerPoint**, **Google Slides**...) : reproductibilité du document (on peut régénérer les slides à partir de leur code source), séparation de l'écriture du contenu et de l'affichage du contenu (système de thème...), cohérence générale des documents générés (automatisation des numéros de pages, des métadonnées du pdf qui respectent la structure du document...). Beamer n'est cependant pas exempt de défauts :

- Le mécanisme de placement d'objets dans un document LaTeX se base sur de l'optimisation mathématique d'un programme avec des contraintes. Ce système est très pratique pour rédiger des documents longs ou complexes, mais est peu adapté à des présentations où le contenu de presque chaque *page* est indépendant des autres pages, et où le contenu de chaque page est souvent très simple.
- Certaines tâches triviales à faire dans un outil WYSIWYG peuvent être très laborieuses, typiquement le positionnement d'images dans une *page* pour en maximiser la visibilité.
- Le coût d'entrée pour écrire ou modifier une présentation Beamer peut être plus élevé qu'avec un outil WYSIWYG.
- L'écosystème de génération de pdf de LaTeX n'est pas *bit à bit reproductible* : même en fixant la chaîne de compilation LaTeX utilisée et le code sources des slides, le fichier pdf généré ne sera pas exactement identique d'une compilation à l'autre.
- Le temps de compilation de *slides* qui comportent de nombreuses pages peut exploser, ce qui nuit à l'ergonomie de l'outil.
- Beamer comporte de nombreuses dépendances, ce qui fait qu'il est presque toujours nécessaire d'avoir la chaîne complète de **TeX Live** pour compiler des *slides*, qui pèse dans les 3 Go.

Objectifs du projet

Le but du projet dans son ensemble est de proposer une alternative légère à Beamer pour en limiter les défauts. L'approche prise sera descriptive plutôt que programmatique. Autrement dit, basée sur un langage de description comme **YAML** plutôt que sur un langage Turing-complet comme **LaTeX**. Ce choix de design limite fortement les possibilités de l'outil, mais permet aussi des optimisations extrêmes lors de la génération de *slides* tout en simplifiant la composition et la réutilisation de *slides* d'une présentation à une autre.

Les objectifs de ce projet sont nombreux et certains de ces objectifs sont listés en page suivante. **Il n'est pas attendu qu'un seul groupe d'étudiant·e·s réalise tous ces objectifs !** Selon le profil des candidat·e·s et leur nombre, des objectifs seront définis pour chaque groupe travaillant sur le projet.

- Définir un format de fichiers descriptif qui permet de représenter la grande majorité des *pages* utilisées dans des *slides*. Ce format devra permettre de définir concisément comment chaque page doit être *mise en page*, ce qui sera fait en s'inspirant des agencements classiques disponibles dans les outils WYSIWYG et des technologies logicielles ayant déjà une abstraction de mise en page (par exemple [Qt Layouts](#)).
- Réaliser une veille des technologies pertinentes pour ce projet (lister des technologies, les évaluer et en réaliser une critique). En particulier sur les outils qui ont un but similaire au nôtre (par exemple [pandoc](#) pour la génération de *slides*), ou sur ceux qui peuvent être utilisés pour résoudre une sous-partie du problème (par exemples [SILE](#) ou [ConTeXt](#) pour la transformation de texte en pdf).
- Implémenter des prototypes d'implémentation en se basant sur les technologies jugées pertinentes par la veille technologique. En particulier, comparer une implémentation se servant d'un langage à la TeX léger comme [ConTeXt](#) contre une implémentation de génération directe des pdf serait intéressant. Permettre la compilation modulaire des slides sera également intéressant à évaluer, par exemple en compilant indépendamment le contenu de chaque *page* puis en composant le document final grâce à un système de build efficace comme [ninja](#).
- Si nécessaire/pertinent, faire des contributions dans les outils open source liés à la génération de pdf ([Ghostscript](#)...) pour en améliorer la reproductibilité bit-à-bit.