# Model de conversion EAST Reveal.js

## Introduction

La conversion entre EAST et Reveal.js se fait principalement par la transposition du XML en HTML.

Mais plusieurs points nécessitent un développement particulier pour assurer leur réalisation.

Ce document se consacre à ces points et défini la démarche à suivre.

## Paramètres globaux

Le paramétrage de EAST se fait à travers une balise XML.

Le paramétrage Reveal.js se fait en javascript.

Barre\_boutons -> controls

Commentaires -> ShowNotes

### Defaults

Certains paramètres doivent être modifiés par défaut pour approcher le fonctionnement de EAST sous Reveal.js

* SlideNumber = true : Les slides sont affichés en bas à droite au lieu du milieu de EAST.
* History = true : On garde l’historique des slides dans la navigation.
* Impression : toujours active.

Certains paramètres doivent différer pour assurer un fonctionnement optimal

mouseWheel = false : Perturbe le fonctionnement du plugin de sommaire.

## Paramètres CSS

Ne sont listés uniquement les éléments nécessitant une attention particulière lors de l’implémentation.

#### PUCE

Level : style du ul>li \* niveau spécifié.

## Style

Couleurs : aucune implémentation n’est faite pour les couleurs sous Reveal. Il est nécessaire de faire des styles CSS.

## Code

La mise en page du font/texte devrait être abandonnée à cause de la coloration automatique par le plugin, plus agréable à l’œil.

## Listes

EAST pousse à faire des liste longues par page du à la taille par défaut de sa police.

Reveal.js a une approche plus moderne : minimum d’idées par slide.

Il est donc préférable de dérouler les listes et de les mettre chacune dans une sous-slide. Les sous-slides permettent de préciser une idée et sont optionnels.

Le déroulement peut se faire d’après la structure suivante :

ul titre liste >li – texte >ul>li>li

=>

slide > titre liste (titre liste) > ul > li | sous slide > titre sous slide (texte) > ul > li >li

## Les formules

L’interpréteur de Reveal est anglais il est possible d’interpréter la formule d’équation, mais uniquement en les écrivant d’après les conventions de latex.

Sinon il est possible d’insérer une image prise ailleurs comme fait EAST (non souhaité).

A PRECISER

## Titre de partie

Les titres de parties n’existent pas sur Reveal.js. Ils sont facilement remplaçables par une slide supplémentaire avec un titre H1 dedans.

## Titre section

Les titres de sections peuvent être en H2

## Liens Internes

Les liens internes doivent être sous la forme #/{n}

## Titre Slide / Liste

Les titres slide / sous slide / liste peuvent entre en H3.

## Logos et les notes de page

Leur implémentation nécessite du code particulier. Ils ne sont pas disponibles par défaut sous Reveal.js.

## Animations

### Transitions entre les slides

Les transitions entre les slides sont portables telles qu’elles sur Reveal.js.

Mais ce n’est pas souhaité : elles ne sont pas optimisées pour le format Reveal, et possèdent souvent les équivalences dans ce logiciel.

|  |  |
| --- | --- |
| **EAST** | **Reveal.js** |

|  |  |
| --- | --- |
| None | None |
| Crossfade  fadeThrough | fade |
| Slide-in | slide |
| Carousel | convex |
| Toss | concave |
| Burst | zoom |

### Animations de listes

Les animations de Reveal.js ne sont pas limités aux listes mais sont globales pour tous les éléments de la présentation.

Les animations d’affichage de liste ne peuvent pas être gardées au click. Cela coupe la présentation. L’implémentation par défaut de ce genre d’animations sur Reveal.js l’intègre dans le flux de la présentation.

C’est ce fonctionnement, qui correspond au fonctionnement des logiciels de présentation lourds, qu’on va garder.

#### Accordéon

La liste accordéon peut se faire avec les classes « fragment current-visible » sur chaque élément qu’on souhaite voir apparaître.

#### Pliage

La liste pliage n’est pas implémentée telle quelle sur Reveal.js.

Un bon remplacement est les classes « fragment grow ». Tous les éléments serons visibles à l’écran et l’élément actuel grandi.

#### Incrémental

La liste incrémentale se fait avec la classe fragment sur chaque élément qu’on souhaite voir apparaître.

#### Incrémental\_ombre

Aucune implémentation disponible pour le moment sur Reveal. Il est possible de la coder. Pour l’instant l’implémentation de Incrémental\_allume sera utilisée à la place avec la couleur bleue.

#### Incrémental\_allume

La liste incrémentale allumée se fait via les classes « fragment highlight-red ». La couleur peut changer contrairement à EAST. Ces classes doivent être ajoutées sur chaque élément qu ‘on veut inclure dans l’animation.

#### Progessif

Même procédé que la liste incrémentale avec l’attribut data-autoslide= « 1000 » en plus sur la liste.

## Plugins additionnels

Notes.js : pour les commentaires du présentateur.

Highlight.js : pour les morceaux de code.

Math.js : pour la mise en page des formules.

Presentable.js : pour le sommaire.