

Maître du temps sous BEAMER

Contrôler l'affichage des différents éléments d'une diapo
Deuxième partie les commandes *overlay*

Bertrand Masson

Les fiches de Bébert

3 mai 2011

On a étudié dans une première fiche l'action de la commande `\pause` qui permet d'afficher les éléments d'une diapo les uns après les autres. Cette commande est très facile d'emploi mais peu flexible. Dans cette fiche nous allons aller plus loin avec les commandes *overlay*

Un point de vocabulaire

Dans cette fiche quand je parle de diapo, je fais référence à l'entité définie par `\begin{frame}...\end{frame}`.

J'emploie le terme de calque (slide) pour faire référence à toutes ces pages créées par des commandes comme `\pause`, qui sont finalement qu'une série de calques qui sont superposés pour fabriquer la diapo. Donc une diapo porte un numéro de page unique et peut être constituée de un ou plusieurs calques.

Il existe plusieurs commandes de type *overlay*, mais elles ont toutes la caractéristique de comprendre l'élément suivant $\langle n \rangle$ où n indique le ou les calques sur lesquels doit apparaître l'élément. n peut prendre les valeurs suivantes :

- $\langle \text{un nombre} \rangle$ par exemple $\langle 2 \rangle$ qui signifie que l'élément concerné n'apparaîtra que sur le calque 2 et sur aucun autre ;
- $\langle n_1, n_2, \dots \rangle$ par exemple $\langle 2, 5, 6 \rangle$ l'élément sera présent uniquement sur les calques 2, 5 et 6 ;
- $\langle n - \rangle$ par exemple $\langle 2 - \rangle$ l'élément sera présent sur le calque 2 et tous les suivants ;
- $\langle -n \rangle$ par exemple $\langle -4 \rangle$ l'élément sera présent uniquement sur les premiers calques et jusqu'au 4^e (c'est à dire les calques 1, 2, 3 et 4) ;
- $\langle n_1 - n_2 \rangle$ par exemple $\langle 3 - 6 \rangle$ l'élément sera présent sur les calques 3 à 6 (c'est à dire les calques 3, 4, 5 et 6) ;

Voici un exemple un peu plus compliqué repris du manuel de BEAMER :
 $\langle -3, 6 - 8, 10, 12 - 15 \rangle$ l'élément apparaît sur les calques : 1, 2, 3, 6, 7, 8, 10, 12, 13, 14, et 15.

BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un `<n>`.

- `\textbf`
- `\textit`
- `\textsl`
- `\includegraphics`
- `\textrm`
- `\textsf`
- `\color`
- `\item`
- `\alert`
- `\structure`

Voici un exemple avec `\color` et `\textbf`

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```

BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un `<n>`.

- `\textbf`
- `\textit`
- `\textsl`
- `\includegraphics`
- `\textrm`
- `\textsf`
- `\color`
- `\item`
- `\alert`
- `\structure`

Voici un exemple avec `\color` et `\textbf`

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```

BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un `<n>`.

- `\textbf`
- `\textit`
- `\textsl`
- `\includegraphics`
- `\textrm`
- `\textsf`
- `\color`
- `\item`
- `\alert`
- `\structure`

Voici un exemple avec `\color` et `\textbf`

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```

BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un `<n>`.

- `\textbf`
- `\textit`
- `\textsl`
- `\includegraphics`
- `\textrm`
- `\textsf`
- `\color`
- `\item`
- `\alert`
- `\structure`

Voici un exemple avec `\color` et `\textbf`

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```

BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un $\langle n \rangle$.

- \textbf
- \textit
- \textsl
- \includegraphics
- \textrm
- \textsf
- \color
- \item
- \alert
- \structure

Voici un exemple avec \color et \textbf

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```


BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un `<n>`.

- `\textbf`
- `\textit`
- `\textsl`
- `\includegraphics`
- `\textrm`
- `\textsf`
- `\color`
- `\item`
- `\alert`
- `\structure`

Voici un exemple avec `\color` et `\textbf`

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```

BEAMER reconnaît certaines commandes de \LaTeX comme pouvant directement être « temporisable ». Il suffit juste de leur adjoindre un `<n>`.

- `\textbf`
- `\textit`
- `\textsl`
- `\includegraphics`
- `\textrm`
- `\textsf`
- `\color`
- `\item`
- `\alert`
- `\structure`

Voici un exemple avec `\color` et `\textbf`

Le code :

```
{\color<2,5->{green}Voici un \textbf<4,6>{exemple}avec \color et \textbf}
```

Till Tantau étant le créateur de BEAMER et Tikz-PGF des commandes de ce dernier sont également « temporisable » directement. Nous les étudierons dans une autre fiche.

Voici un autre exemple avec une liste :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<3> Deuxième item ;
\item<2-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;

Voici un autre exemple avec une liste :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<3> Deuxième item ;
\item<2-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;

- Troisième item ;

Voici un autre exemple avec une liste :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<3> Deuxième item ;
\item<2-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;
- Deuxième item ;
- Troisième item ;

Voici un autre exemple avec une liste :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<3> Deuxième item ;
\item<2-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;

- Troisième item ;
- Dernier item.

Puisque l'on est dans les listes restons-y. Il y a plusieurs manières d'obtenir le même résultat. Prenons l'exemple suivant :

Le code :

```
\begin{itemize}
\item<1-> Premier item;
\item<2-> Deuxième item;
\item<3-> Troisième item;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item;

Puisque l'on est dans les listes restons-y. Il y a plusieurs manières d'obtenir le même résultat. Prenons l'exemple suivant :

Le code :

```
\begin{itemize}
\item<1-> Premier item;
\item<2-> Deuxième item;
\item<3-> Troisième item;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item;
- Deuxième item;

Puisque l'on est dans les listes restons-y. Il y a plusieurs manières d'obtenir le même résultat. Prenons l'exemple suivant :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<2-> Deuxième item ;
\item<3-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;
- Deuxième item ;
- Troisième item ;

Puisque l'on est dans les listes restons-y. Il y a plusieurs manières d'obtenir le même résultat. Prenons l'exemple suivant :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<2-> Deuxième item ;
\item<3-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;
- Deuxième item ;
- Troisième item ;
- Dernier item.

Puisque l'on est dans les listes restons-y. Il y a plusieurs manières d'obtenir le même résultat. Prenons l'exemple suivant :

Le code :

```
\begin{itemize}
\item<1-> Premier item ;
\item<2-> Deuxième item ;
\item<3-> Troisième item ;
\item<4-> Dernier item.
\end{itemize}
```

Voici le résultat :

- Premier item ;
- Deuxième item ;
- Troisième item ;
- Dernier item.

Bien entendu ceci (et les commandes suivantes) est valable pour tous les types de listes (*enumerate* et *description*)

Pour éviter d'avoir à décaler ta numérotation si tu rajoutes un élément au début de ta liste, tu peux utiliser la notation suivante, si le défilement doit ce faire dans l'ordre.

Le code :

```
\begin{itemize}
\item<+--> Premier item;
\item<+--> Deuxième item;
\item<+--> Troisième item;
\item<+--> Dernier item.
\end{itemize}
```

Tu peux encore simplifier en utilisant la forme suivante :

Le code :

```
\begin{itemize}[<+-->]
\item Premier item;
\item Deuxième item;
\item Troisième item;
\item Dernier item.
\end{itemize}
```

Pour mettre, lors de la « temporisation », le texte en « alert » (de couleur rouge par défaut).

Le code :

```
\begin{enumerate}
\item<alert@3> Premier item ;
\item<alert@2> Deuxième item ;
\item<alert@1> Troisième item ;
\item<alert@4> Dernier item.
\end{enumerate}
```

- Premier item ;
- Deuxième item ;
- **Troisième item ;**
- Dernier item.

Le chiffre après le @ indique sur quel calque intervient la mise en « alert ». Tu peux utiliser des formes plus complexes comme <alert@2-> ou <alert@2-4,6>.

Pour mettre, lors de la « temporisation », le texte en « alert » (de couleur rouge par défaut).

Le code :

```
\begin{enumerate}
\item<alert@3> Premier item ;
\item<alert@2> Deuxième item ;
\item<alert@1> Troisième item ;
\item<alert@4> Dernier item.
\end{enumerate}
```

- Premier item ;
- **Deuxième item ;**
- Troisième item ;
- Dernier item.

Le chiffre après le @ indique sur quel calque intervient la mise en « alert ». Tu peux utiliser des formes plus complexes comme <alert@2-> ou <alert@2-4,6>.

Pour mettre, lors de la « temporisation », le texte en « alert » (de couleur rouge par défaut).

Le code :

```
\begin{enumerate}
\item<alert@3> Premier item ;
\item<alert@2> Deuxième item ;
\item<alert@1> Troisième item ;
\item<alert@4> Dernier item.
\end{enumerate}
```

- Premier item ;
- Deuxième item ;
- Troisième item ;
- Dernier item.

Le chiffre après le @ indique sur quel calque intervient la mise en « alert ». Tu peux utiliser des formes plus complexes comme <alert@2-> ou <alert@2-4,6>.

Pour mettre, lors de la « temporisation », le texte en « alert » (de couleur rouge par défaut).

Le code :

```
\begin{enumerate}
\item<alert@3> Premier item ;
\item<alert@2> Deuxième item ;
\item<alert@1> Troisième item ;
\item<alert@4> Dernier item.
\end{enumerate}
```

- Premier item ;
- Deuxième item ;
- Troisième item ;
- **Dernier item.**

Le chiffre après le @ indique sur quel calque intervient la mise en « alert ». Tu peux utiliser des formes plus complexes comme <alert@2-> ou <alert@2-4,6>.

Si le défilement doit se faire dans l'ordre tu peux écrire :

Le code :

```
\begin{enumerate}
\item<alert@+> Premier item;
\item<alert@+> Deuxième item;
\item<alert@+> Troisième item;
\item<alert@+> Dernier item.
\end{enumerate}
```

Simplifiable en

Le code :

```
\begin{enumerate}[<alert@+>]
\item Premier item;
\item Deuxième item;
\item Troisième item;
\item Dernier item.
\end{enumerate}
```

Tu peux combiner l'apparition et la mise en « alert ».

Le code :

```
\begin{enumerate}
\item<3-|alert@3> Premier item;
\item<2-|alert@2> Deuxième item;
\item<1-|alert@1> Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- Troisième item;

Tu peux combiner l'apparition et la mise en « alert ».

Le code :

```
\begin{enumerate}
\item<3-|alert@3> Premier item;
\item<2-|alert@2> Deuxième item;
\item<1-|alert@1> Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- Deuxième item;
- Troisième item;

Tu peux combiner l'apparition et la mise en « alert ».

Le code :

```
\begin{enumerate}
\item<3-|alert@3> Premier item;
\item<2-|alert@2> Deuxième item;
\item<1-|alert@1> Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- Premier item;
- Deuxième item;
- Troisième item;

Tu peux combiner l'apparition et la mise en « alert ».

Le code :

```
\begin{enumerate}
\item<3-|alert@3> Premier item;
\item<2-|alert@2> Deuxième item;
\item<1-|alert@1> Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- Premier item;
- Deuxième item;
- Troisième item;
- **Dernier item.**

Si le défilement doit se faire dans l'ordre tu peux écrire :

Le code :

```
\begin{enumerate}
\item<+ -|alert@+> Premier item;
\item<+ -|alert@+> Deuxième item;
\item<+ -|alert@+> Troisième item;
\item<+ -|alert@+> Dernier item.
\end{enumerate}
```

Simplifiable en

Le code :

```
\begin{enumerate}[<+ -|alert@+>]
\item Premier item;
\item Deuxième item;
\item Troisième item;
\item Dernier item.
\end{enumerate}
```

Tu peux même changer l'aspect du marqueur.

Le code :

```
\begin{enumerate}
\item<1-|alert@1>[A.] Premier item;
\item<2-|alert@2> Deuxième item;
\item<3-|alert@3>[\ding{52}] Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

A. Premier item;

Tu peux même changer l'aspect du marqueur.

Le code :

```
\begin{enumerate}
\item<1-|alert@1>[A.] Premier item;
\item<2-|alert@2> Deuxième item;
\item<3-|alert@3>[\ding{52}] Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

A. Premier item;

- Deuxième item;

Tu peux même changer l'aspect du marqueur.

Le code :

```
\begin{enumerate}
\item<1-|alert@1>[A.] Premier item;
\item<2-|alert@2> Deuxième item;
\item<3-|alert@3>[\ding{52}] Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- A. Premier item;
- Deuxième item;
- ✓ Troisième item;

Tu peux même changer l'aspect du marqueur.

Le code :

```
\begin{enumerate}
\item<1-|alert@1>[A.] Premier item;
\item<2-|alert@2> Deuxième item;
\item<3-|alert@3>[\ding{52}] Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- A. Premier item;
- Deuxième item;
- ✓ Troisième item;
- Dernier item.

Tu peux même changer l'aspect du marqueur.

Le code :

```
\begin{enumerate}
\item<1-|alert@1>[A.] Premier item;
\item<2-|alert@2> Deuxième item;
\item<3-|alert@3>[\ding{52}] Troisième item;
\item<4-|alert@4> Dernier item.
\end{enumerate}
```

- A. Premier item;
- Deuxième item;
- ✓ Troisième item;
- Dernier item.

`\ding` permet d'afficher les symboles *Zapf dingbats* en utilisant le package *pifont* (voir la fiche sur les listes).

Pour « temporiser » du texte en dehors de liste ou de commande directement « temporisable », on va utiliser les deux commandes `\only` et `\onslide`. Elles s'utilisent de la même façon :

Le code :

```
\only<n>{le texte à temporiser}  
\onslide<n>{le texte à temporiser}  
\only<n> le texte à temporiser...  
\onslide<n> le texte à temporiser...
```

Si tu n'utilises pas les accolades, tout le texte suivant la commande et ce jusqu'à la fin de la diapo sera « temporisé ».

Le `<n>` prend les mêmes valeurs que dans les exemples précédents. La différence entre les deux commandes, est que `\onslide` contrairement à `\only` réserve l'espace occupé par le texte affecté à la « temporisation ».

Un petit exemple explicatif. Pour bien visualiser ce qui se passe j'ai mis en bleu le texte « temporisé ».

Le code :

Un morceau de phrase avec `\onslide<2>\textcolor{blue}{un`
`texte à temporiser}}`, la suite de la phrase.

Un morceau de phrase avec `\only<2>\textcolor{blue}{un`
`texte à temporiser}}`, la suite de la phrase.

Un morceau de phrase avec _____ , la suite de la
phrase.

Un morceau de phrase avec _____ , la suite de la phrase.

Un petit exemple explicatif. Pour bien visualiser ce qui se passe j'ai mis en bleu le texte « temporisé ».

Le code :

Un morceau de phrase avec `\onslide<2>{\textcolor{blue}{un texte à temporiser}}`, la suite de la phrase.

Un morceau de phrase avec `\only<2>{\textcolor{blue}{un texte à temporiser}}`, la suite de la phrase.

Un morceau de phrase avec `un texte à temporiser`, la suite de la phrase.

Un morceau de phrase avec `un texte à temporiser`, la suite de la phrase.

Avant de continuer avec les différentes commandes de temporisation je vais évoquer rapidement la transparence sous BEAMER. Par défaut les objets absents d'un calque sont invisibles. Pour pouvoir apprécier les subtilités des différentes commandes de BEAMER, j'ai remplacé cette invisibilité par une transparence de 30 % par la commande `\setbeamercovered{transparent=30}`. Je ne rentre pas plus dans les détails ce sera l'objet d'une autre fiche.

Maintenant je vais pouvoir aborder les deux commandes suivantes `\visible` et `\uncover`.

Avec `\visible` les objets non présents sur un calque sont invisibles.

Avec `\uncover` ils sont transparents avec la valeur définie par `\setbeamercovered{transparent=30}`.

Un exemple sur la diapo suivante :

Le code :

Un morceau de phrase avec `\visible<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec `\uncover<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec _____, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Le code :

Un morceau de phrase avec `\visible<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec `\uncover<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Si je reprends ma commande `\onslide` avec ma transparence à 30 %, tu remarques, ci-dessous qu'elle se comporte comme `\uncover` :

Le code :

Un morceau de phrase avec `\onslide<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Si je reprends ma commande `\onslide` avec ma transparence à 30 %, tu remarques, ci-dessous qu'elle se comporte comme `\uncover` :

Le code :

```
Un morceau de phrase avec \onslide<2>{un texte à  
temporiser}, la suite de la phrase.
```

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Si je reprends ma commande `\onslide` avec ma transparence à 30 %, tu remarques, ci-dessous qu'elle se comporte comme `\uncover` :

Le code :

```
Un morceau de phrase avec \onslide<2>{un texte à temporiser}, la suite de la phrase.
```

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Il existe deux marqueurs pour modifier le comportement de `\onslide` + et *. Avec + `\onslide` se comporte comme `\visible` et avec * il se comporte comme `\only`. Des exemples diapo suivante :

Le code :

Un morceau de phrase avec `\onslide<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec `\onslide+<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec `\onslide*<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Un morceau de phrase avec , la suite de la phrase.

Un morceau de phrase avec , la suite de la phrase.

Le code :

Un morceau de phrase avec `\onslide<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec `\onslide+<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec `\onslide*<2>`{un texte à temporiser}, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Voici d'autres commandes :

`\invisible<n>{texte}` : c'est le contraire de `\visible` , les objets sont invisibles sur les calques précisés.

`\alt<n>{nouveau texte}{texte par défaut}`, remplace le texte par défaut par un autre texte sur les calques spécifiés.

Le code :

Un morceau de phrase avec `\invisible<2>{un texte à temporiser}`, la suite de la phrase.

Un morceau de phrase avec `\alt<2>{bi bo bu sur le calque 2}{ma ta sa sur les autres calques}`, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Un morceau de phrase avec ma ta sa sur les autres calques, la suite de la phrase.

Voici d'autres commandes :

`\invisible<n>{texte}` : c'est le contraire de `\visible` , les objets sont invisibles sur les calques précisés.

`\alt<n>{nouveau texte}{texte par défaut}` , remplace le texte par défaut par un autre texte sur les calques spécifiés.

Le code :

Un morceau de phrase avec `\invisible<2>{un texte à temporiser}` , la suite de la phrase.

Un morceau de phrase avec `\alt<2>{bi bo bu sur le calque 2}{ma ta sa sur les autres calques}` , la suite de la phrase.

Un morceau de phrase avec _____ , la suite de la phrase.

Un morceau de phrase avec bi bo bu sur le calque 2, la suite de la phrase.

Voici d'autres commandes :

`\invisible<n>{texte}` : c'est le contraire de `\visible` , les objets sont invisibles sur les calques précisés.

`\alt<n>{nouveau texte}{texte par défaut}`, remplace le texte par défaut par un autre texte sur les calques spécifiés.

Le code :

Un morceau de phrase avec `\invisible<2>{un texte à temporiser}`, la suite de la phrase.

Un morceau de phrase avec `\alt<2>{bi bo bu sur le calque 2}{ma ta sa sur les autres calques}`, la suite de la phrase.

Un morceau de phrase avec un texte à temporiser, la suite de la phrase.

Un morceau de phrase avec ma ta sa sur les autres calques, la suite de la phrase.

Tu peux utiliser cette commande pour changer l'aspect d'un mot, le mettre en gras ou en couleur par exemple :

`\alt<2>{\textbf{mot}}{mot}`.

On va passer un peu plus de temps sur la commande `\temporal`. Cette commande permet d'alterner 3 textes différents.

```
\temporal<n>{un texte avant <n>}{le texte pendant <n>}{un texte après <n>}
```

Le code :

```
Un morceau de phrase avec \temporal<2>{crac sur le calque  
1}{boum sur le calque 2}{hue sur le calaque 3}, les fill's  
en tomb'nt à mes g'noux.
```

Un morceau de phrase avec crac sur le calque 1, les fill's en tomb'nt à mes g'noux.

On va passer un peu plus de temps sur la commande `\temporal`. Cette commande permet d'alterner 3 textes différents.

```
\temporal<n>{un texte avant <n>}{le texte pendant <n>}{un texte après <n>}
```

Le code :

```
Un morceau de phrase avec \temporal<2>{crac sur le calque  
1}{boum sur le calque 2}{hue sur le calaque 3}, les fill's  
en tomb'nt à mes g'noux.
```

Un morceau de phrase avec boum sur le calque 2, les fill's en tomb'nt à mes g'noux.

On va passer un peu plus de temps sur la commande `\temporal`. Cette commande permet d'alterner 3 textes différents.

```
\temporal<n>{un texte avant <n>}{le texte pendant <n>}{un texte après <n>}
```

Le code :

```
Un morceau de phrase avec \temporal<2>{crac sur le calque  
1}{boum sur le calque 2}{hue sur le calque 3}, les fill's  
en tomb'nt à mes g'noux.
```

Un morceau de phrase avec hue sur le calque 3, les fill's en tomb'nt à mes g'noux.

Attention pour voir apparaître « hue sur le calque 3 » il faut qu'il y ait un minimum de 3 calques. Il faut donc ajouter en fin de diapo la commande `\onslide+<3>`.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec crac sur les calques 1 et 2, les fill's en
tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec crac sur les calques 1 et 2, les fill's en
tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec boum sur les calques 3, 4 et 5, les fill's en tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec boum sur les calques 3, 4 et 5, les fill's en tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec boum sur les calques 3, 4 et 5, les fill's en tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec hue sur le calque 6 et les suivants, les fill's en tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec hue sur le calque 6 et les suivants, les fill's en tomb'nt à mes g'noux.

Des exemples un peu plus complexes :

Le code :

```
Un morceau de phrase avec \temporal<3-5>{crac sur les  
calques 1 et 2}{boum sur les calques 3, 4 et 5}{hue sur la  
calques 6 et les suivantes}, les fill's en tomb'nt à mes  
g'noux.
```

Un morceau de phrase avec hue sur le calque 6 et les suivants, les fill's en tomb'nt à mes g'noux.

Attention j'ai ajouté \onslide+<8> pour créer 8 calques.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec crac sur les calques 1, 2 et 4, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec crac sur les calques 1, 2 et 4, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec boum sur les calques 3 et 5, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec crac sur les calques 1, 2 et 4, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec boum sur les calques 3 et 5, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec hue sur le calque 6 et les suivantes, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec hue sur le calque 6 et les suivantes, les fill's en tomb'nt à mes g'noux.

Le code :

Un morceau de phrase avec `\temporal<3,5>`{crac sur les calques 1, 2 et 4}{boum sur les calques 3 et 5}{hue sur le calque 6 et les suivantes}, les fill's en tomb'nt à mes g'noux.

Un morceau de phrase avec hue sur le calque 6 et les suivantes, les fill's en tomb'nt à mes g'noux.

Attention j'ai ajouté `\onslide+<8>` pour créer 8 calques.

L'un des intérêt de cette commande est de pouvoir changer la couleur du texte de 3 façons différentes. Dans l'exemple suivant le texte va être en gris sur les premiers calques quand je n'ai pas encore abordé le sujet, puis en rouge quand j'évoque le sujet et enfin en noir j'en ai terminé avec le sujet.

Le code :

```
Sur le calque 1 le \temporal<2>{\textcolor{gray!50}{sujet}}{\textcolor{red}{sujet}}{\textcolor{}}sujet}}est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon \temporal<4>{\textcolor{gray!50}{exemple}}{\textcolor{red}{exemple}}{\textcolor{black}{exemple}}qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.
```

Sur le calque 1 le `sujet` est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon `exemple` qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

L'un des intérêt de cette commande est de pouvoir changer la couleur du texte de 3 façons différentes. Dans l'exemple suivant le texte va être en gris sur les premiers calques quand je n'ai pas encore abordé le sujet, puis en rouge quand j'évoque le sujet et enfin en noir j'en ai terminé avec le sujet.

Le code :

```
Sur le calque 1 le \temporal<2>{\textcolor{gray!50}{sujet}}{\textcolor{red}{sujet}}{\textcolor{}}sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon \temporal<4>{\textcolor{gray!50}{exemple}}{\textcolor{red}{exemple}}{\textcolor{black}{exemple}} qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.
```

Sur le calque 1 le **sujet** est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon **exemple** qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

L'un des intérêt de cette commande est de pouvoir changer la couleur du texte de 3 façons différentes. Dans l'exemple suivant le texte va être en gris sur les premiers calques quand je n'ai pas encore abordé le sujet, puis en rouge quand j'évoque le sujet et enfin en noir j'en ai terminé avec le sujet.

Le code :

```
Sur le calque 1 le \temporal<2>{\textcolor{gray!50}{sujet}}{\textcolor{red}{sujet}}{\textcolor{}}sujet}est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon \temporal<4>{\textcolor{gray!50}{exemple}}{\textcolor{red}{exemple}}{\textcolor{black}{exemple}}qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.
```

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon exemple qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

L'un des intérêt de cette commande est de pouvoir changer la couleur du texte de 3 façons différentes. Dans l'exemple suivant le texte va être en gris sur les premiers calques quand je n'ai pas encore abordé le sujet, puis en rouge quand j'évoque le sujet et enfin en noir j'en ai terminé avec le sujet.

Le code :

```
Sur le calque 1 le \temporal<2>{\textcolor{gray!50}{sujet}}{\textcolor{red}{sujet}}{\textcolor{}}sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon \temporal<4>{\textcolor{gray!50}{exemple}}{\textcolor{red}{exemple}}{\textcolor{black}{exemple}} qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.
```

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon **exemple** qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

L'un des intérêt de cette commande est de pouvoir changer la couleur du texte de 3 façons différentes. Dans l'exemple suivant le texte va être en gris sur les premiers calques quand je n'ai pas encore abordé le sujet, puis en rouge quand j'évoque le sujet et enfin en noir j'en ai terminé avec le sujet.

Le code :

```
Sur le calque 1 le \temporal<2>{\textcolor{gray!50}{sujet}}{\textcolor{red}{sujet}}{\textcolor{}}sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon \temporal<4>{\textcolor{gray!50}{exemple}}{\textcolor{red}{exemple}}{\textcolor{black}{exemple}} qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.
```

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon exemple qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

Si tu utilises plusieurs fois ce principe dans ta présentation, il peut être judicieux de créer une commande. Je vais te montrer comment faire car il y a quelques subtilités. Tout d'abord on ne peut pas utiliser `\newcommand` pour créer notre nouvelle commande car `\newcommand` n'aime pas les `< >` nécessaires pour définir `\temporal`. On va donc utiliser la commande `\def` qui est une commande de \TeX . Ensuite on va utiliser différemment `\temporal`. Tu te souviens que si on n'utilise pas le `{}`, dans `\onslide` par exemple, c'est tout le texte qui suit qui est affecté par la commande. On va utiliser le même principe ici en réduisant sa portée par des `{}` si nécessaire. On va créer notre commande `\couleur` par le code suivant que tu places dans l'entête de ton document :

Le code :

```
\def\couleur<#1>{\temporal<#1>{\color{gray!50}}
{\color{red}}{\color{black}}}
```

Commande à écrire sur une seule ligne.

La diapo suivante te montre comment utiliser la nouvelle commande.

Le code :

```
\begin{itemize}
\couleur<1> \item premier item;
\couleur<2> \item second item;
\couleur<3> \item troisième item;
\couleur<4> \item dernier item.
\end{itemize}
```

Sur le calque 1 le `{\couleur<2> sujet}` est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon `{\couleur<4> exemple}` qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

Le résultat page suivante.

- premier item ;
- second item ;
- troisième item ;
- dernier item.

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon exemple qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

- premier item ;
- **second item ;**
- troisième item ;
- dernier item.

Sur le calque 1 le **sujet** est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon **exemple** qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

- premier item ;
- second item ;
- troisième item ;
- dernier item.

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon exemple qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

- premier item ;
- second item ;
- troisième item ;
- **dernier item.**

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon **exemple** qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

- premier item ;
- second item ;
- troisième item ;
- dernier item.

Sur le calque 1 le sujet est en gris, il passe en rouge sur le deuxième, enfin il est en noir sur le troisième. Mon exemple qui jusqu'ici était en gris passe en rouge sur le calque 4 et passe en noir sur le cinquième.

Les blocks sont directement temporisables.

Le code :

```
\begin{block}<3->{Le titre}
```

Le texte

```
\end{block}
```

```
\begin{exampleblock}<2->{Le titre}
```

Le texte

```
\end{exampleblock}
```

Les blocks sont directement temporisables.

Le code :

```
\begin{block}<3->{Le titre}
```

Le texte

```
\end{block}
```

```
\begin{exampleblock}<2->{Le titre}
```

Le texte

```
\end{exampleblock}
```

Le titre

Le texte

Les blocks sont directement temporisables.

Le code :

```
\begin{block}<3->{Le titre}
```

Le texte

```
\end{block}
```

```
\begin{exampleblock}<2->{Le titre}
```

Le texte

```
\end{exampleblock}
```

Le titre

Le texte

Le titre

Le texte

À chaque commande de temporisation que nous avons vu, il existe un environnement qui se comporte de la même façon. `\only` ⇒ `onlyenv`, `\alt` ⇒ `altenv`, `\visible` ⇒ `visibleenv`, `\uncover` ⇒ `uncoverenv`, and `\invisible` ⇒ `invisibleenv`. Par exemple :

Le code :

```
\begin{invisibleenv}<n>  
le texte  
\end{invisibleenv}
```

Le comportement de *altenv* est toutefois un peu différent. Voici l'exemple du manuel :

Le code :

```
Ce  
\begin{altenv}<2>{( )}{[ ]}  
mot  
\end{altenv}
```

est entouré de parenthèse sur le calque 2 et de crochet sur le calque 1.

Ce [mot] est entouré de parenthèse sur le calque 2 et de crochet sur le calque 1.

Le comportement de *altenv* est toutefois un peu différent. Voici l'exemple du manuel :

Le code :

```
Ce  
\begin{altenv}<2>{( )}{[ ]}  
mot  
\end{altenv}
```

est entouré de parenthèse sur le calque 2 et de crochet sur le calque 1.

Ce (mot) est entouré de parenthèse sur le calque 2 et de crochet sur le calque 1.

Quand on veut remplacer un texte par un autre ou une image par une autre on est souvent confronté à un problème de décalage de ligne surtout quand les objets ne sont pas de mêmes dimensions. Je vois bien que ce que je raconte n'est pas très clair donc passons à un petit exemple. Voici le code qui régit la diapo suivante :

Le code :

Un petit exemple

```
\includegraphics[scale=0.5]<1>{marmotte.png}  
\includegraphics[scale=0.5]<2>{marmotte2.png}
```

qui montre le problème du décalage, peu esthétique, des lignes entre le premier et le deuxième calque.

Un petit exemple



qui montre le problème du décalage, peu esthétique, des lignes entre le premier et le deuxième calque.

Un petit exemple



qui montre le problème du décalage, peu esthétique, des lignes entre le premier et le deuxième calque.

Pour régler ce problème on va créer une zone de temporisation dont la taille sera supérieure ou égale à l'image la plus grande :

Le code :

Un petit exemple

```
\begin{overlayarea}{0.5\textwidth}{0.6\textheight}  
\includegraphics[scale=0.5]<1>{marmotte.png}  
\includegraphics[scale=0.5]<2>{marmotte2.png}  
\end{overlayarea}
```

qui montre que le problème du décalage des lignes, grâce à *overlayarea* a disparu.

Un petit exemple



qui montre que le problème du décalage des lignes, grâce à `overlayarea` a disparu.

Un petit exemple



qui montre que le problème du décalage des lignes, grâce à `overlayarea` a disparu.

La commande `\overprint` à le même rôle que `\overlayarea`, mais est moins souple car on ne peut contrôler que la largeur de la zone temporisable, par défaut la largeur du texte.

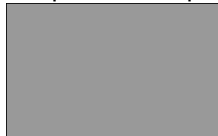
Le code :

Un petit exemple

```
\begin{overprint}
\includegraphics<1>{carre1.png}
\includegraphics<2>{carre2.png}
\end{overprint}
```

qui montre que le problème du décalage des lignes, grâce à *overprint* a disparu.

Un petit exemple



qui montre que le problème du décalage des lignes, grâce à *overprint* a disparu.

La commande `\overprint` à le même rôle que `\overlayarea`, mais est moins souple car on ne peut contrôler que la largeur de la zone temporisable, par défaut la largeur du texte.

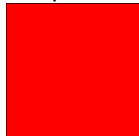
Le code :

Un petit exemple

```
\begin{overprint}  
\includegraphics<1>{carre1.png}  
\includegraphics<2>{carre2.png}  
\end{overprint}
```

qui montre que le problème du décalage des lignes, grâce à *overprint* a disparu.

Un petit exemple



qui montre que le problème du décalage des lignes, grâce à *overprint* a disparu.

Imaginons que sur un fond de carte on veuille faire apparaître différents aménagements, par exemple une future autoroute et ses différentes aires de repos et ses entrées et sorties. Pour alléger le document on pourrait avoir une image de fond et des images ne comportant que les éléments additionnels. Ces images sont symbolisées par les vignettes suivantes,

La diapo suivante te montre comment procéder.

Imaginons que sur un fond de carte on veuille faire apparaître différents aménagements, par exemple une future autoroute et ses différentes aires de repos et ses entrées et sorties. Pour alléger le document on pourrait avoir une image de fond et des images ne comportant que les éléments additionnels. Ces images sont symbolisées par les vignettes suivantes, le rectangle vert = le fond de carte,



La diapo suivante te montre comment procéder.

Imaginons que sur un fond de carte on veuille faire apparaître différents aménagements, par exemple une future autoroute et ses différentes aires de repos et ses entrées et sorties. Pour alléger le document on pourrait avoir une image de fond et des images ne comportant que les éléments additionnels. Ces images sont symbolisées par les vignettes suivantes, le rectangle vert = le fond de carte, le trait noir = la future autoroute,



La diapo suivante te montre comment procéder.

Imaginons que sur un fond de carte on veuille faire apparaître différents aménagements, par exemple une future autoroute et ses différentes aires de repos et ses entrées et sorties. Pour alléger le document on pourrait avoir une image de fond et des images ne comportant que les éléments additionnels. Ces images sont symbolisées par les vignettes suivantes, le rectangle vert = le fond de carte, le trait noir = la future autoroute, les étoiles rouges = les sorties



La diapo suivante te montre comment procéder.

Imaginons que sur un fond de carte on veuille faire apparaître différents aménagements, par exemple une future autoroute et ses différentes aires de repos et ses entrées et sorties. Pour alléger le document on pourrait avoir une image de fond et des images ne comportant que les éléments additionnels. Ces images sont symbolisées par les vignettes suivantes, le rectangle vert = le fond de carte, le trait noir = la future autoroute, les étoiles rouges = les sorties et les cercles jaunes = les aires de repos.



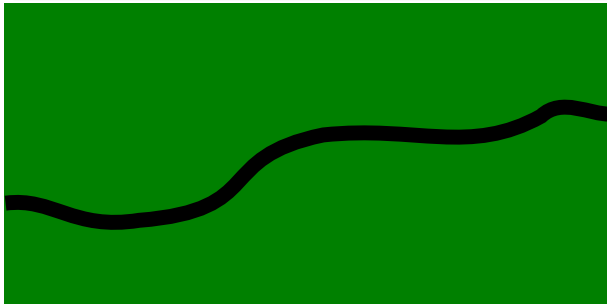
La diapo suivante te montre comment procéder.

Le code :

```
\includegraphics<1->{carte1.pdf}  
\llap{\includegraphics<2->{carte2.pdf}}  
\llap{\includegraphics<3->{carte3.pdf}}  
\llap{\includegraphics<4->{carte4.pdf}}
```

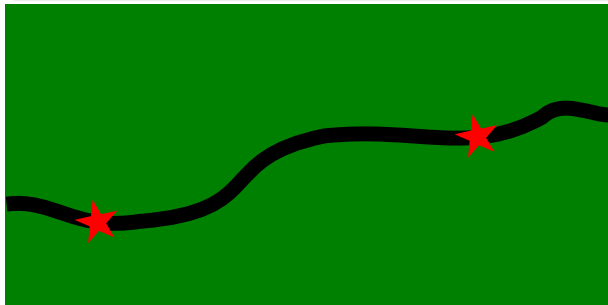
Le code :

```
\includegraphics<1->{carte1.pdf}  
\llap{\includegraphics<2->{carte2.pdf}}  
\llap{\includegraphics<3->{carte3.pdf}}  
\llap{\includegraphics<4->{carte4.pdf}}
```



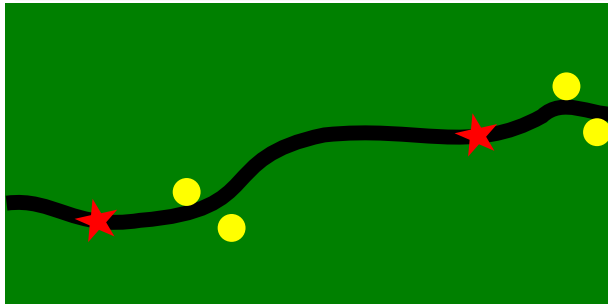
Le code :

```
\includegraphics<1->{carte1.pdf}  
\llap{\includegraphics<2->{carte2.pdf}}  
\llap{\includegraphics<3->{carte3.pdf}}  
\llap{\includegraphics<4->{carte4.pdf}}
```



Le code :

```
\includegraphics<1->{carte1.pdf}  
\llap{\includegraphics<2->{carte2.pdf}}  
\llap{\includegraphics<3->{carte3.pdf}}  
\llap{\includegraphics<4->{carte4.pdf}}
```



On a vu dans la fiche consacrée à la commande `\pause` comment faire apparaître progressivement les lignes d'un tableau. Pour faire de même avec les colonnes il faut utiliser `\onslide` :

Le code :

```
\rowcolors[ ]{1}{blue!20}{blue!10}
\begin{tabular}{l!{\vrule}c<{\onslide<2->}c<{\onslide<3->}c<{\onslide<4->}c<{\onslide}}
Tableau & A & B & C & D \\
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6 \\
Z & 5 & 6 & 7 & 8
\end{tabular}
```

Attention au traitement de la dernière colonne où l'on a juste `\onslide`, car `\onslide<5->` à un comportement étrange.

Tableau	A
X	1
Y	3
Z	5

On a vu dans la fiche consacrée à la commande `\pause` comment faire apparaître progressivement les lignes d'un tableau. Pour faire de même avec les colonnes il faut utiliser `\onslide` :

Le code :

```
\rowcolors[ ]{1}{blue!20}{blue!10}
\begin{tabular}{l!{\vrule}c<{\onslide<2->}c<{\onslide<3->}c<{\onslide<4->}c<{\onslide}}
Tableau & A & B & C & D \\
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6 \\
Z & 5 & 6 & 7 & 8
\end{tabular}
```

Attention au traitement de la dernière colonne où l'on a juste `\onslide`, car `\onslide<5->` à un comportement étrange.

Tableau	A	B
X	1	2
Y	3	4
Z	5	6

On a vu dans la fiche consacrée à la commande `\pause` comment faire apparaître progressivement les lignes d'un tableau. Pour faire de même avec les colonnes il faut utiliser `\onslide` :

Le code :

```
\rowcolors[ ]{1}{blue!20}{blue!10}
\begin{tabular}{l!{\vrule}c<{\onslide<2->}c<{\onslide<3->}c<{\onslide<4->}c<{\onslide}}
Tableau & A & B & C & D \\
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6 \\
Z & 5 & 6 & 7 & 8
\end{tabular}
```

Attention au traitement de la dernière colonne où l'on a juste `\onslide`, car `\onslide<5->` à un comportement étrange.

Tableau	A	B	C
X	1	2	3
Y	3	4	5
Z	5	6	7

On a vu dans la fiche consacrée à la commande `\pause` comment faire apparaître progressivement les lignes d'un tableau. Pour faire de même avec les colonnes il faut utiliser `\onslide` :

Le code :

```
\rowcolors[ ]{1}{blue!20}{blue!10}
\begin{tabular}{l!{\vrule}c<{\onslide<2->}c<{\onslide<3->}c<{\onslide<4->}c<{\onslide}}
Tableau & A & B & C & D \\
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6 \\
Z & 5 & 6 & 7 & 8
\end{tabular}
```

Attention au traitement de la dernière colonne où l'on a juste `\onslide`, car `\onslide<5->` à un comportement étrange.

Tableau	A	B	C	D
X	1	2	3	4
Y	3	4	5	6
Z	5	6	7	8

Voilà, on a fait le tour des commandes de temporisation dans BEAMER. Dans les prochaines fiches consacrées à BEAMER on verra comment coupler TikZ avec BEAMER pour ajouter de l'inter-action dans les dessins.