

sim-os-menus [fr]

Présenter un 'terminal', une 'visionneuse' ou un 'menu contextuel' comme sur un ordi.

Version 0.1.1 -- 14/11/2023

Cédric Pierquet

c pierquet -- at -- outlook . fr

<https://github.com/cpierquet/sim-os-menus>

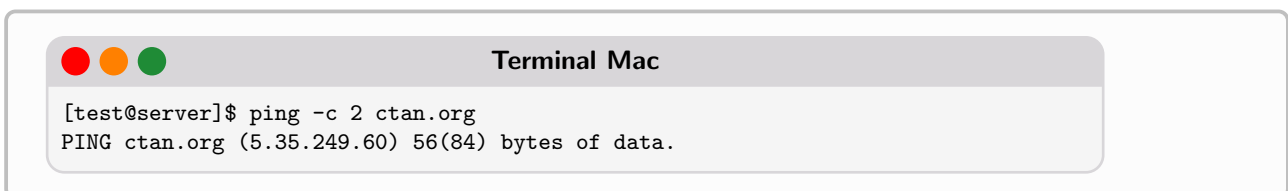
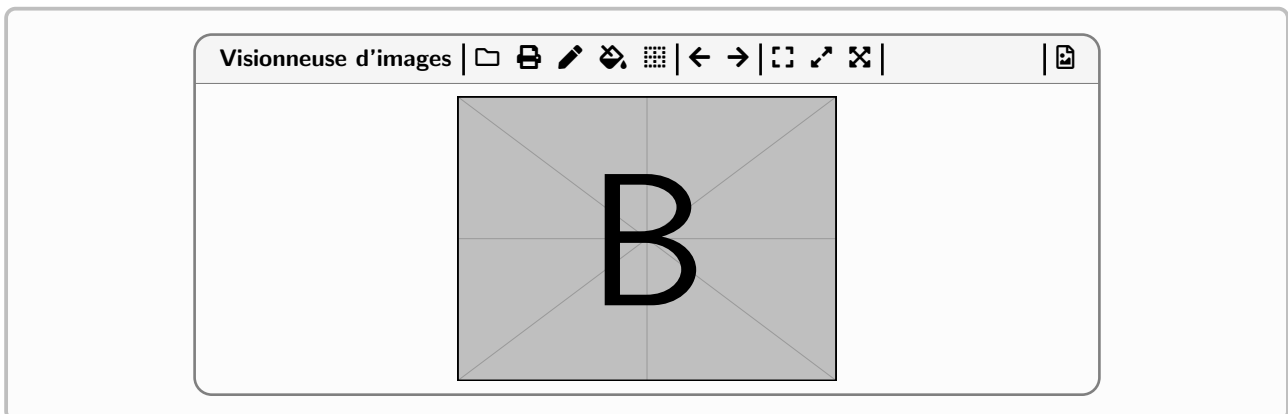
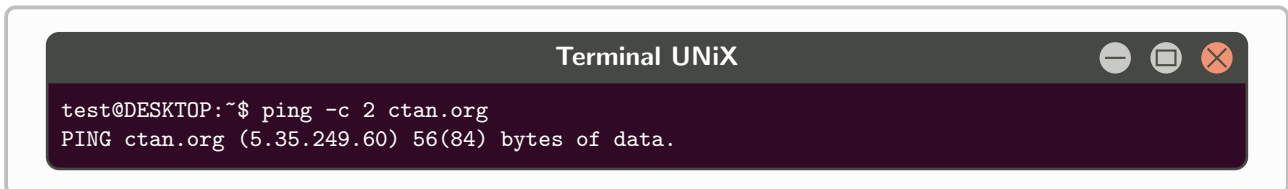
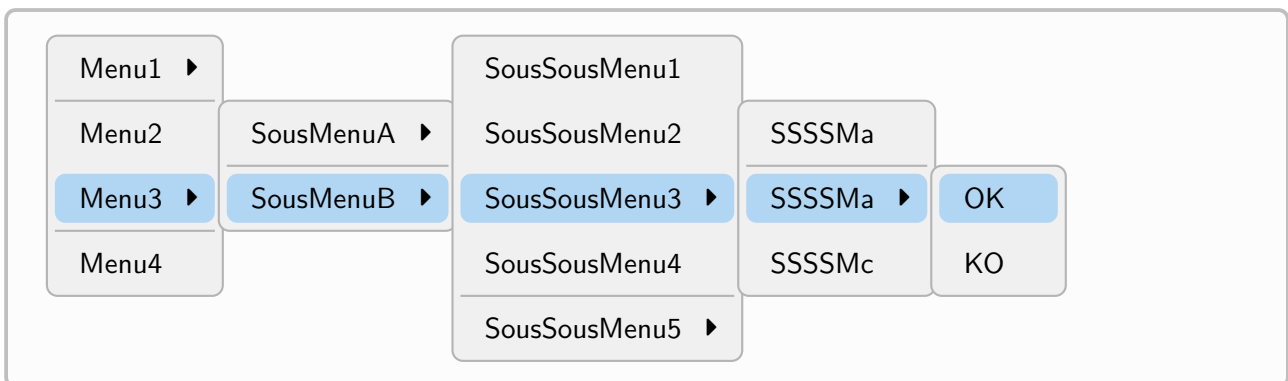


Table des matières

1	Introduction	2
1.1	Description	2
1.2	Chargement	2
1.3	Historique	2
2	Les commandes disponibles	3
2.1	Menu contextuel	3
2.2	Terminal	5
2.3	Viewers	7

1 Introduction

1.1 Description

Avec ce packages, il est possible de créer un « faux » terminal, ou un « faux » menu contextuel, ou encore une « fausse » visionneuse, comme sur un écran d'ordinateur !

Les styles sont *globalement* fixés, mais quelques personnalisations sont possibles.

1.2 Chargement

Pour charger le package, il suffit d'ajouter dans le préambule :

```
\usepackage{sim-os-menus}
```

Les packages utiles, et chargés sont :

- `tikz` (avec `calc,positioning`), `pgf`, `pgffor`;
- `calc`, `fontawesome5`;
- `simplekv`, `xintexpr`, `listofitems`, `xstring`;
- `settobox`, `tabularray`;
- `tcolorbox` (avec `breakable,fitting,skins,listings,listingsutf8,hooks`).

1.3 Historique

0.1.1 : Ajout des commandes en [fr]

0.1.0 : Version initiale

2 Les commandes disponibles

2.1 Menu contextuel

Dans le but de créer un menu contextuel, la commande est :

```
%----menu contextuel
\MenuContextuel[clés]{liste des items}<options tikz>
```

Les clés, optionnelles et entre [...], sont :

- `CoulFond` := couleur du fond;
- `CoulSurlig` := couleur du surlignage;
- `Arrondi` := booléen pour arrondir les coins (`true` par défaut);
- `Police` := police pour les items (`\normalsize\normalfont` par défaut);
- `CoulItems` := couleur(s) des items (`black` par défaut);
- `MargeV` := marge verticale des lignes (`6pt` par défaut);
- `MargeH` := marge horizontale des lignes (`12pt` par défaut);
- `Fleche` := caractère pour la flèche (`\faCaretRight` par défaut);
- `ListeSeps` := liste des éventuels séparateurs (vide ou pour tous les niveaux!);
- `ListeIcones` := liste des éventuelles icones (vide ou pour tous les niveaux/items!);
- `ListeDecals` := liste des éventuels décalages verticaux pour les sous-menus (à partir du n°2, ...!) (vide ou pour tous les sous-niveaux!);
- `Icones` := booléen pour afficher les icones (`false` par défaut);
- `Barre` := booléen pour une petite barre verticale après les éventuelles icones `small vertical bar with icons` (`true` par défaut);
- `Space` := espacement horizontal entre les sous-menus (`-0.125` par défaut).

L'argument obligatoire, entre {...}, correspond à la liste des items, sous la forme :

```
item1A,item1B,... § item2A,item2B,... § ...
```

- si un item se termine par `(*)`, cela correspond au *début* du sous-menu suivant (un seul par niveau!);
- si un item se termine par `(>)` (avant un éventuel `(*)`), une *flèche* est écrite à la fin de la ligne de l'item.

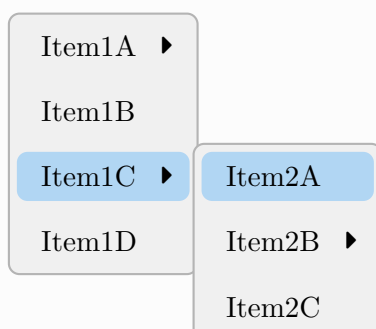
Une syntaxe correcte est fondamentale pour que le code présente correctement le menu contextuel!

Quelques conseils, du fait des clés sensibles `ListeIcones`, `ListeDecals` et `ListeSeps` :

- `ListeIcones` doit avoir le même nombre d'éléments que le nombre de menus/items (avec des items vides possibles);
- `ListeSeps` doit avoir le même nombre d'éléments que le nombre de niveaux (avec des niveaux vides possibles);
- `ListeDecals` doit avoir le même nombre d'éléments que le nombre de sous-menus (Avec 0 si pas de décalage!).

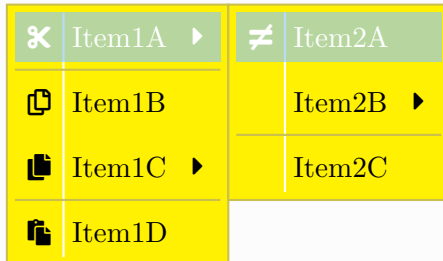
%style par défaut

```
\MenuContextuel{Item1A(>),Item1B,Item1C(>)(*),Item1D § Item2A(*),Item2B(>),Item2C}
```

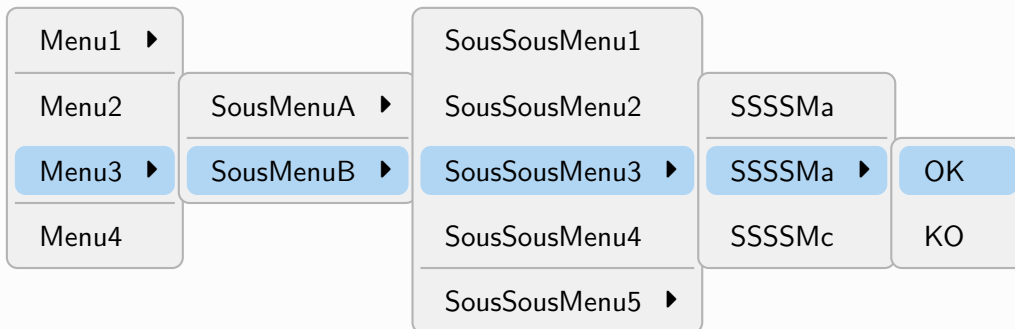


%style personnalis 

```
\MenuContextuel[Arrondi=false,CoulFond=yellow,CoulSurlign=teal,%  
ListeSeps={1,3/2},CoulItems={black/white},Icones,Espacement=0,%  
ListeIcones={\faCut,\faIcon[regular]{copy},\faCopy,\faPaste / \faNotEqual}]  
{Item1A(>)(*),Item1B,Item1C(>),Item1D § Item2A(*),Item2B(>),Item2C}
```



```
\MenuContextuel[Police=\sffamily,ListeSeps={1,3/1/4/1/},ListeDecals={1,2,1,0}]{%  
Menu1(>),Menu2,Menu3(>)(*),Menu4 §  
SousMenuA(>),SousMenuB(>)(*) §  
SousSousMenu1,SousSousMenu2,SousSousMenu3(>)(*),SousSousMenu4,SousSousMenu5(>) §  
SSSSMa,SSSSMa(>)(*),SSSSMc §  
OK(*),KO  
}
```



2.2 Terminal

Pour créer un « faux » terminal (Win/UNiX/Mac), les environnements sont :

```
%----Terminal Windows
\begin{TerminalWin}[clés]{options tcolorbox}
...
\end{TerminalWin}

%----Terminal UNiX
\begin{TerminalUnix}[clés]{options tcolorbox}
...
\end{TerminalUnix}

%----Terminal OSX
\begin{TerminalMac}[clés]{options tcolorbox}
...
\end{TerminalMac}
```

Les clés, optionnelles et entre [...], sont :

- **Titre** := titre du terminal (Terminal Win/UNiX/Mac par défaut);
- **Align** := alignement horizontal (paragraphe) du terminal (**center** par défaut);
- **Largeur** := largeur du terminal (`\linewidth` par défaut);
- **Icones** := booléen pour afficher des icones (**true** par défaut).

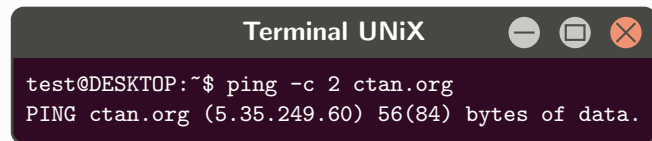
L'argument obligatoire, entre {...}, correspond à des options spécifiques, à passer à la `tcolorbox`.

```
\begin{TerminalWin}{}
Microsoft Windows [version 10.0.22000.493]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\test>ping ctan.org
Envoi d'une requête 'ping' sur ctan.org [5.35.249.60] avec 32 octets de données :
Réponse de 5.35.249.60 : octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60 : octets=32 temps=37 ms TTL=51
Réponse de 5.35.249.60 : octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60 : octets=32 temps=39 ms TTL=51
Statistiques Ping pour 5.35.249.60:
Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
Minimum = 35ms, Maximum = 39ms, Moyenne = 36ms
\end{TerminalWin}
```

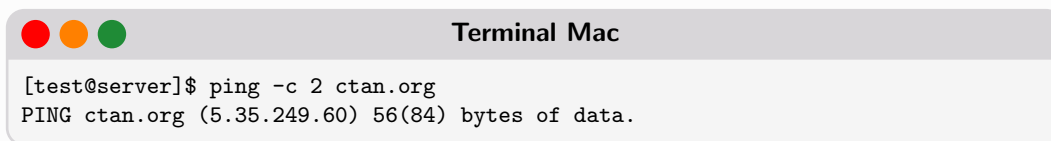
>_ Terminal Win

```
Microsoft Windows [version 10.0.22000.493]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\test>ping ctan.org
Envoi d'une requête 'ping' sur ctan.org [5.35.249.60] avec 32 octets de données :
Réponse de 5.35.249.60 : octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60 : octets=32 temps=37 ms TTL=51
Réponse de 5.35.249.60 : octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60 : octets=32 temps=39 ms TTL=51
Statistiques Ping pour 5.35.249.60:
Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
Minimum = 35ms, Maximum = 39ms, Moyenne = 36ms
```

```
\begin{TerminalUnix}[Align=flush right]{hbox}
test@DESKTOP:~$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
\end{TerminalUnix}
```



```
\begin{TerminalMac}[Largeur=14cm,Align=flush left]{}
[test@server]$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
\end{TerminalMac}
```



2.3 Visionneuses

Pour créer une « fausse » visionneuse, les environnements sont :

```
%---Visionneuse PDF
\begin{VisionnPDF}[clés]{options tcolorbox}
    ....
\end{VisionnPDF}

%---Visionneuse d'images
\begin{VisionnIMG}[clés]{options tcolorbox}
    ....
\end{VisionnIMG}
```

Les clés, optionnelles et entre [...], sont :

- **Titre** := titre du terminal (Terminal Win/UNiX/Mac par défaut) ;
- **Align** := alignement horizontal (paragraphe) de la visionneuse (**center** par défaut) ;
- **AlignH** := alignement horizontal (du contenu) de la visionneuse (**center** par défaut) ;
- **Largeur** := largeur de la visionneuse (`\linewidth` par défaut) ;
- **Icones** := booléen pour afficher des icones (**true** par défaut).

L'argument obligatoire, entre {...}, correspond à des options spécifiques, à passer à la tcolorbox.

```
\begin{VisionnPDF}{hbox}
\fbbox{\includegraphics [page=35,width=6cm]{ProfLycee-doc.pdf}}%
\fbbox{\includegraphics [page=36,width=6cm]{ProfLycee-doc.pdf}}%
\end{VisionnPDF}
```

```
\begin{VisionnPDF}{hbox}
\fbbox{\includegraphics [page=35,width=6cm]{ProfLycee-doc.pdf}}%
\fbbox{\includegraphics [page=36,width=6cm]{ProfLycee-doc.pdf}}%
\end{VisionnPDF}
```

Visionneuse PDF | | | | | | | |

10 Suites récurrentes et « toile »

10.1 Idée

? **Idée(s)**

L'idée est d'obtenir une commande pour tracer (en TikZ) la « toile » permettant d'obtenir – graphiquement – les termes d'une suite récurrente définie par une relation $u_{n+1} = f(u_n)$.
Comme pour les autres commandes TikZ, l'idée est de laisser l'utilisateur définir et créer son environnement TikZ, et d'insérer la commande `\tikzrecurse` pour afficher la « toile ».

10.2 Commandes

Code LaTeX

```
...
\begin{tikzpicture}[options]
...
\tikzrecurse[clic]{options du tracé}[options supplémentaires des termes]
...
\end{tikzpicture}
```

Clés et options

Plusieurs (arguments) (optionnels) sont disponibles :

- la clé **(Fct)** qui définit la fonction f ;
- la clé **(Nom)** qui est le nom de la suite ;
- la clé **(No)** qui est l'indice initial ;
- la clé **(U0)** qui est la valeur du terme initial ;
- la clé **(Nb)** qui est le nombre de termes à construire ;
- la clé **(PostLabel)** qui est le placement des labels par rapport à l'axe (Ox) ;
- la clé **(DecalLabel)** qui correspond au décalage des labels par rapport aux abscisses ;
- la clé **(TailleLabel)** qui correspond à la taille des labels ;
- un booléen **(AffTermes)** qui permet d'afficher les termes de la suite sur l'axe (Ox) ;
- le deuxième argument optionnel concerne les (options) du tracé de l'outil en langage TikZ ;
- le troisième argument optionnel concerne les (options) du tracé des termes en langage TikZ ;

Information(s)

Il est à noter que le code n'est pas autonome, et doit être intégré dans un environnement `\tikzpicture`.
L'utilisateur est donc libre de définir ses styles pour l'affichage des éléments de son graphique, et il est libre également de rajouter des éléments en plus du tracé de la « toile » !
La macro ne permet – pour le moment – ni de tracer la bissectrice, ni de tracer la courbe...
En effet, il n'y a pas trop d'options pour ces deux éléments, et l'idée est quand même de conserver une commande simple ! Donc l'utilisateur se chargera de tracer et de personnaliser sa courbe et sa bissectrice !

10.3 Exemples

Information(s)

On va tracer la toile des 4 premiers termes de la suite récurrente
$$\begin{cases} u_1 = 1 \\ u_{n+1} = \sqrt{3u_n} + 1 \text{ pour tout entier } n \geq 1 \end{cases}$$

Code LaTeX

```
Code file
\def\x{1.5cm}\def\y{1.5cm}
\def\lmax{0}\def\lmax2{0}\def\lgrille{1}\def\lgrille2{0.5}
\def\ymin{0}\def\ymax{8}\def\ygrille{1}\def\ygrille2{0.5}
\draw[step=\lgrille,ystep=\ygrille,linewidth=0.5pt,lightgray!50] (\xmin,\ymin) grid
-- (\xmax,\ymax);
\draw[linewidth=1.5pt,--darkgray,--latex] (\xmin,0)--(\xmax,0);
\draw[linewidth=1.5pt,--darkgray,--latex] (0,\ymin)--(0,\ymax);
\foreach \x in {0,1,...,8} {\draw[darkgray,linewidth=0.5pt] (\x,0pt) -- (\x,-0.4pt)};
\foreach \y in {0,1,...,7} {\draw[darkgray,linewidth=1.5pt] (0pt,\y) -- (-0.4pt,\y)};
\draw[thick,ForestGreen,domain=0:8,samples=2] plot (\x,\f(\x));
\draw[thick,ForestGreen,domain=0:8,samples=2] plot (\x,\x);
```

Information(s)

Peut être que – tabulièrement – des options localisées seront disponibles pour un tracé géométrique de la courbe et de la bissectrice, mais pour le moment la macro ne fait que l'outil.

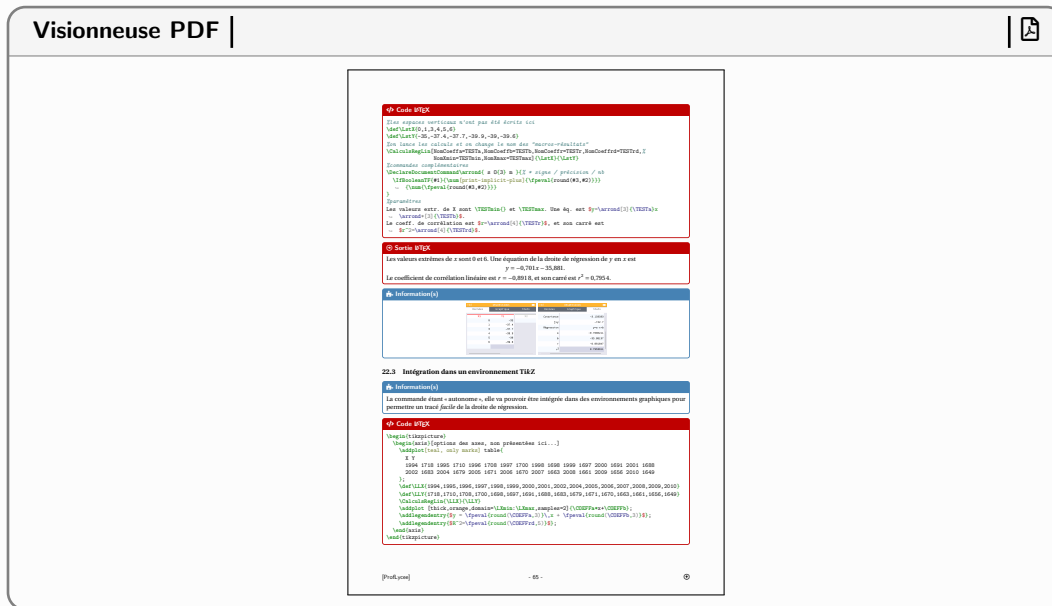
[sim-os-menus]

- 7 -

```

\begin{VisionnPDF}[Largeur=14cm,Icones=false]{ }
\fbbox{\includegraphics [page=65,width=4.75cm]{ProfLycee-doc.pdf}}
\end{VisionnPDF}

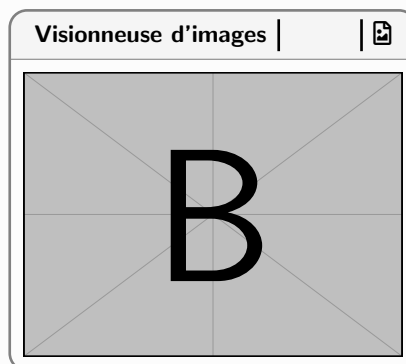
```



```

\begin{VisionnIMG}[Icones=false]{hbox}
\includegraphics [width=5cm]{example-image-b}
\end{VisionnIMG}

```



```

\begin{VisionnIMG}[Largeur=12cm]{ }
\includegraphics [width=3cm]{example-image}
\end{VisionnIMG}

```

