

Résumé

Abstract

<https://plmlab.math.cnrs.fr/mchupin/mpchess>

Table des matières

1	Installation	2
1.1	Avec la T _E Xlive sous Linux ou MacOS	2
1.2	Avec MikT _E X et Windows	3
1.3	Dépendances	3
1.4	Utilisation avec LuaT _E X et luamplib	3
2	Pourquoi ce package et philosophie générale	3
3	Plateau	4
3.1	Réglage des tailles	5
3.2	Nombre de case	5
3.3	Dimension d'une case	6
3.4	Réglage du thème de couleur	6
3.5	Affichage des coordonnées	7
3.6	Vue blanche ou noire	7
3.7	Noms des joueurs	7
4	Pièces et positions	7
4.1	Réglage du thème des pièces	7
4.2	Lecture de données au format PGN	7
4.3	Lecture de données au format FEN	7
5	Annotation	7

1 Installation

MPchess est sur le CTAN et peut être installé via le gestionnaire de package de votre distribution.

<https://www.ctan.org/pkg/mpchess>

1.1 Avec la T_EXlive sous Linux ou MacOS

Pour installer MPchess avec T_EXlive, il vous faudra créer le répertoire texmf dans votre home.

```
user $> mkdir ~/texmf
```

Ensuite, il faudra y placer les fichiers .mp dans le répertoire

~/texmf/tex/metapost/mpchess/

MPchess est constitué de 5 fichiers METAPOST :

- mpchess.mp;
- mpchess-chessboard.mp;
- mpchess-pgn.mp;
- mpchess-fen.mp;
- mpchess-cburnett.mp.

Une fois fait cela, MPchess sera chargé avec le classique

```
input mpchess
```

1.2 Avec MikTeX et Windows

Ces deux systèmes sont inconnus de l'auteur de MPchess, ainsi, nous renvoyons à leurs documentations pour y ajouter des packages locaux :

<http://docs.miktex.org/manual/localadditions.html>

1.3 Dépendances

MPchess dépend des packages METAPOST : **hatching** et, si MPchess n'est pas utilisé avec Lua \TeX et **luamplib**, **latexmp**.

1.4 Utilisation avec Lua \TeX et **luamplib**

Il est tout à fait possible d'utiliser MPchess directement dans un fichier \TeX avec Lua \TeX et le package **luamplib**. C'est d'ailleurs ce qui est fait pour écrire cette documentation.

MPchess utilise, pour certaines fonctionnalités, l'opérateur **infont** de METAPOST. Ainsi, pour que le contenu de ces fonctionnalités soient composé dans la fonte courante du document, on devra ajouter dans son document \TeX , la commande :

```
\mplibtexttextlabel{enable}
```

Pour plus de détails sur ces mécanismes, nous renvoyons à la documentation du package **luamplib** [3].

2 Pourquoi ce package et philosophie générale

Il existe déjà des packages \TeX pour dessiner des plateaux d'échecs et des positions dont le très bon **xskak** [2] qui couplé avec le package **chessboard** [1]. Ulrike Fisher a réalisé là un travail d'amélioration, de maintien, et nous a fournit

d'excellents outils pour réaliser des diagrammes d'échecs et de traiter les différents formats de descriptions de parties¹. Les documentations de ces packages sont de très bonnes qualités.

Plusieurs choses ont motivé la création de *MPchess*. Tout d'abord, avec *chessboard* l'ajout d'ensemble de pièces n'est pas très aisé car cela repose sur des fontes. De plus, je trouve que le dessin de diagrammes de parties d'échec est quelque chose de très graphique, et que le passage par un langage dédié au dessin offre plus de souplesse.

Avec *MPchess*, on construit l'image finale du plateau d'échec avec les pièces par couches successives. Ainsi, on commencera par produire et dessiner le plateau (*backboard*), que l'on pourra modifier en colorant par exemple certaines cases, ensuite on ajoutera les pièces de la position (*chessboard*), et enfin, on pourra annoter le tout avec des marques, des couleurs, des flèches, etc.

Par ailleurs, *MPchess* produit des images proches graphiquement de ce que peut fournir l'excellent site *open source* <https://lichess.org>. Vous verrez que les couleurs, les pièces, et l'aspect général sont largement inspirés de ce que propose ce site.

3 Plateau

Le plateau est appelé avec *MPchess* *backboard*. Il faudra initialiser le plateau avant de le dessiner. Cela se fait avec la commande suivante :

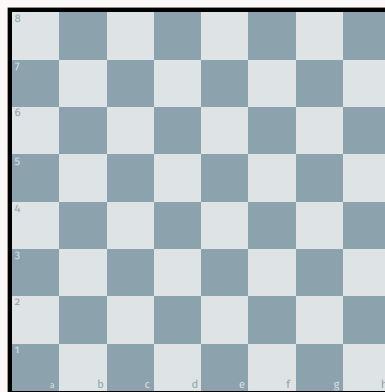
init_backboard

Cette commande construit une *picture* de METAPOST nommée *backboard*. Il faudra ensuite la tracer comme l'illustre l'exemple suivant.

Exemple 1

```
input mpchess

beginfig(0);
init_backboard;
draw backboard;
endfig;
```



Cette initialisation permettra de prendre en compte les différentes options et fonctionnalités que nous allons décrire dans la suite.

1. Elle a même développé le package *chessfss* pour gérer divers fontes d'échec.

3.1 Réglage des tailles

Lors de la création du `backboard`, on peut décider de la largeur de celui-ci. Cela se fait grâce à la commande suivante :

`set_backboard_width(<dim>)`

<dim> : est la largeur de plateau de jeu souhaitée (avec l'unité). Par défaut, cette dimension est à 5 cm.

L'utilisation de cette commande est illustré à l'exemple 2. Cette commande est à utilisée avant `init_backboard` pour qu'elle soit prise en compte à la création de l'image.

On peut récupérer la dimension du plateau de jeu par la commande suivante.

`get_backboard_width`

Cette commande retourne un type `numeric`.

3.2 Nombre de case

Par défaut, le plateau de jeu contient 64 cases (8×8). On peut modifier cela avec la commande suivante :

`set_backboard_size(<nbr>)`

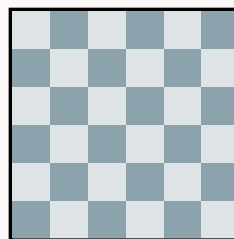
<nbr> : est le nombre de cases souhaité. Le plateau sera alors carré de taille $\langle nbr \rangle \times \langle nbr \rangle$. Par défaut ce nombre est à 8.

Encore une fois, cette commande est à utilisée avant `init_backboard` pour qu'elle soit prise en compte comme le montre l'exemple suivant.

Exemple 2

```
input mpchess

beginfig(0);
set_backboard_width(3cm);
set_backboard_size(6);
init_backboard;
draw backboard;
endfig;
```



Pour obtenir la taille du plateau de jeu par la commande suivante.

`get_backboard_size`

Cette commande retourne un type `numeric`.

3.3 Dimension d'une case

En fonction du nombre de cases sur le plateau et la largeur prescrite pour le plateau, `MPchess` calcule la dimension (largeur ou hauteur) d'une case. Cela sert d'unité générale. Pour l'obtenir, on utilisera la commande suivante.

`get_square_dim`

Cette commande retourne un `numeric`.

3.4 Réglage du thème de couleur

Plusieurs thèmes de couleurs sont accessibles. Pour choisir un thème de couleur, on utilisera la commande suivante :

`set_color_theme(<string>)`

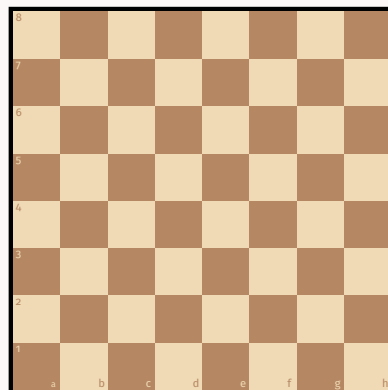
où `<string>` peut valoir :

- `"BlueLichess"` (thème par défaut);
- `"BrownLichess"`;
- ou `"Classical"`.

Les exemples montrent les résultats obtenus.

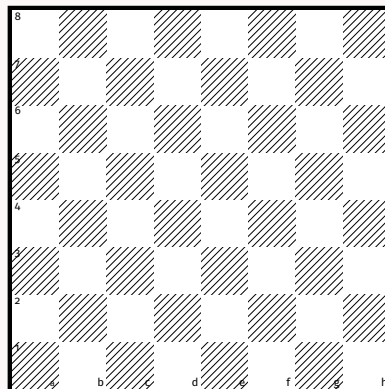
Exemple 3

```
input mpchess
beginfig(0);
set_color_theme("
    BrownLichess");
init_backboard;
draw backboard;
endfig;
```



Exemple 4

```
input mpchess
beginfig(0);
set_color_theme("Classical
");
init_backboard;
draw backboard;
endfig;
```



Les deux thèmes colorés fournis sont les couleurs des thèmes de Lichess.

3.5 Affichage des coordonnées

3.6 Vue blanche ou noire

3.7 Noms des joueurs

4 Pièces et positions

4.1 Réglage du thème des pièces

4.2 Lecture de données au format PGN

4.3 Lecture de données au format FEN

5 Annotation

```
test_initbackboard(test)
```

Références

- [1] Ulrike FISCHER. *The chessboard package. Print chess boards.* Version 1.9. 9 mars 2023. URL : <https://ctan.org/pkg/chessboard>.
- [2] Ulrike FISCHER. *The xskak package. An extension to the skak package for chess typesetting.* Version 1.5. 9 mars 2023. URL : <https://ctan.org/pkg/xskak>.
- [3] Hans HAGEN et al. *The luamplib package. Use LuaTeX's built-in MetaPost interpreter.* Version 2.23.0. 12 jan. 2022. URL : <https://ctan.org/pkg/luamplib>.

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<code>get_backboard_size</code>	<code>get_square_dim,</code>	<i>6</i>	<code>,</code>	<i>5</i>
<code>,</code>	<i>5</i>	<code>init_backboard,</code>	<i>4</i>	<code>set_backboard_width</code>
<code>get_backboard_width</code>			<code>,</code>	<i>5</i>
<code>,</code>	<i>5</i>	<code>set_backboard_size</code>	<code>set_color_theme,</code>	<i>6</i>