



## Résumé

Abstract

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

## Table des matières

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

## 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<sub>E</sub>Xlive sous Linux ou MacOS

Pour installer MPchess avec T<sub>E</sub>Xlive, 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** [**luamplib**].

## 2 Philosophie générale

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 :

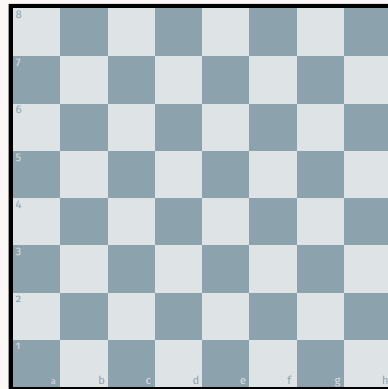
`init_backboard`

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.

#### 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 ( $8 \times 8$ ). On peut modifier cela avec la commande suivante :

`set_backboard_size(<nbr>)`

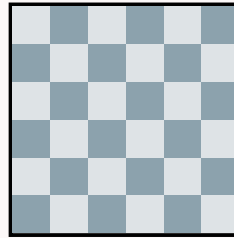
**<nbr>** : est le nombre de cases souhaité. Le plateau sera alors carré de taille  $\langle nbr \rangle \times \langle nbr \rangle$ . Par défaut ce nombre est à 8.

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

### 3.4 Réglage du thème de couleur

### 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>`