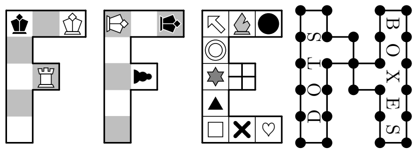
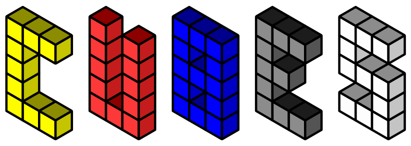


ffen2tex v0.9

Alain Brobecker, 2012-2015



What is ffen2tex?

ffen2tex is a pre-processor to create **diagrams with square grids** or **cubes setups** in the TEX typesetting system. For example you can make diagrams for Chess, Checkers, Draughts, Go, fairy variants of those games and for puzzles...
The name comes from FEN (Forsyth-Edwards Notation) which was created to describe Chess positions (position=diagram+who has the move+castling rights+etc...). In 2002, Joost de Heer designed the **Fairy FEN** extension for fairy Chess diagrams (not positions!). I added some more possibilities for my own purpose. You will have more informations in the examples or by searching "Forsyth-Edwards Notation" on the internet.

How to use it: pre-processing

The following explanations were written for someone using a window$/m$do$ system. I hope this example is enough to understand how to use ffen2tex and for people using another operating system.

Instead of adding commands to TEX i chose to make a program that must be executed on your .tex file before you process it with TEX (so it must be pre-processed by ffen2tex). It converts the %FFEN instructions into pstricks command, so at first you need to have the standard **pstricks package installed**. You must also **copy ffen2tex.prg** in your TEX work directory.

Now create a file named "sample.tex" in your TEX work directory and write the following in it:

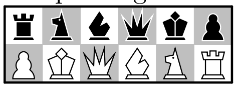
```
\documentclass{article}
\usepackage{pstricks}
\begin{document}
Sample diagram:\\
%FFEN rnbqkp/PKQBNR
\end{document}
```

Then under the m$do$ prompt (or the command line of your OS) type the following:

```
c:\tex>ffen2tex sample.tex temporary.tex
c:\tex>latex temporary.tex
c:\tex>dvips temporary.ps
c:\tex>del sample.pdf
c:\tex>rename temporary.pdf sample.pdf
c:\tex>del temporary.*
```

The resulting "sample.pdf" file contains the following:

Sample diagram:



Of course we don't want to type this sequence every time we process a .tex file, so we will use a batch (shell?) file to do it. I saved the following batch sequence in "goffen.bat" which is in my TEX work directory:

```
ffen2tex %1.tex temporary.tex
latex temporary.tex
dvips temporary.dvi
ps2pdf temporary.ps
del %1.pdf
rename temporary.pdf %1.pdf
del temporary.*
```

Then i only have to type this under the m$do$ prompt:

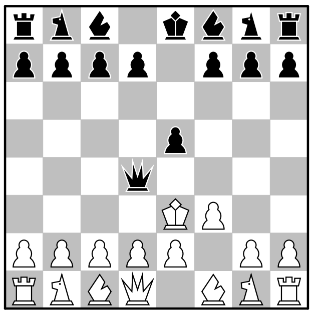
```
c:\tex>goffen sample
```

Some examples are following. The list of all variables, modifiers and pieces is in the Quick reference. Happy diagramming...

ffen2tex v0.9 - %FFEN examples

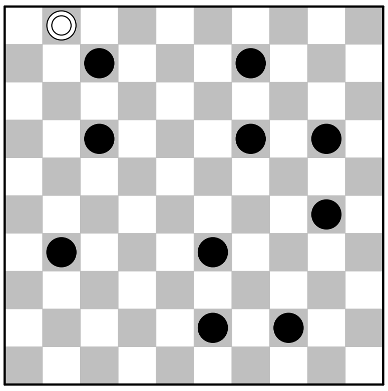
Alain Brobecker, august 2015

Since **ffen2tex** is based upon the Forsyth-Edwards Notation, the default settings are for Chess diagrams. The one below has been reached after black's third move, how did the game went?



%FFEN rnb1kbnr/pppp1ppp/8/4p3/3q4/4KP2/PPPPP1PP/RNBQ1BNR

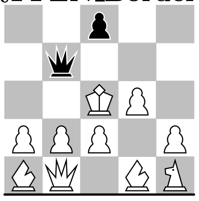
Diagrams can have any size between 1 × 1 and 99 × 99, but for empty areas of width bigger than 9 squares you'll have to use two digits or more. Here a width of 10 empty squares is given as two empty areas of width 5, ie 55, but it could have been 64 or 73... Below is an international Draughts game with FFEN_PieceCount set.



1+10: White plays and wins.

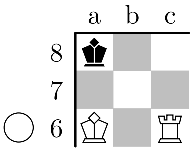
%FFEN_PieceCount 1
%FFEN 1D8/2c3c3/55/2c3c1c1/55/8c1/1c3c4/55/5c1c2/55

We show here a 5 × 5 part of the first example with only the down border. The %FFEN_Fill variable was changed to **1=odd (ie upper left is filled)** to have the correct square coloring. **Note that %FFEN_BorderAll is evaluated before the other %FFEN_Border modifiers.**



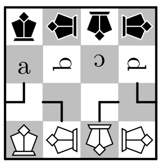
%FFEN_PieceCount 0
%FFEN_Fill 1
%FFEN_BorderAll 0
%FFEN_BorderDown 1
%FFEN 2p2/1q3/2KP1/PPP1P/BQ1BN

To put coordinates we insert letters and figures with the ' escape character (double ' for text with two characters), and put them outside the board using the # modifier. It's tedious but flexible. **Note that %FFEN_Fill is still 1, because the values are not reinitialised between two diagrams.** Upper left corner would be colored if it were not outside the board. I added a white circle indicating who has to move, and yes, it's mate in 1.



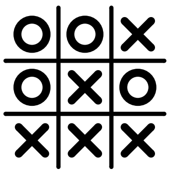
%FFEN_BorderAll 0
%FFEN_BorderUp 1
%FFEN_BorderLeft 1
%FFEN #2#'a#'b#'c/#1#'8k2/#1#'73/#C#'6K1R

With the *N modifier we can rotate the pieces, this is often used for fairy pieces in Chess problems.



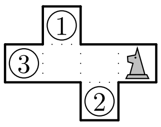
%FFEN_Fill 2
%FFEN_BorderAll 1
%FFEN k*1k*2k*3k/'a*1'b*2'c*3'd/v*1v*2v*3v/K*1K*2K*3K

By modifying the size, the inner lines, the borders and the filling, we leave the world of Chess/Draughts/Checkers for Tic-tac-toe. What was the last move here?



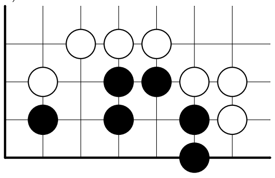
```
%FFEN_SquareSize 0.7
%FFEN_Fill 0
%FFEN_InnerStyle 1
%FFEN_InnerWidth 0.08
%FFEN_BorderStyle 0
%FFEN ..x/.x./xxx
```

White pieces can be colored with the - or = modifiers. Pieces can be circled with the @ modifier. And using the # modifier we can make odd-shaped boards (it can also apply on many empty squares).



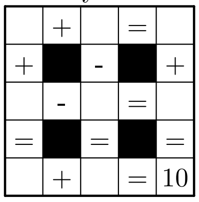
```
%FFEN_SquareSize 0.5
%FFEN_InnerStyle 3          %FFEN_InnerWidth 0.03
%FFEN_BorderStyle 1
%FFEN #1@'1#2/@'32-N/#2@'2#1
```

With the ∧ and >modifiers it's possible to put pieces inbetween squares. This is used in Chess for some retrograde analysis problems, but the most obvious use is for Go (even better: use the **igo** package by Étienne Dupuis). Can white kill here?



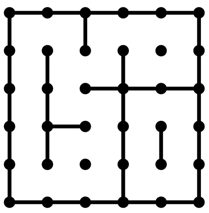
```
%FFEN_BorderRight 0
%FFEN_BorderUp 0
%FFEN_InnerStyle 1
%FFEN_InnerWidth 0.015
%FFEN 7/1∧>C∧>C∧>C3/∧>C1∧>c∧>c∧>C∧>C1/∧>c1∧>c1∧>c∧>C1/#4#∧>c#2
```

The $ modifier changes the fill property of a particular square (it can also apply on many empty squares and to outside squares). Here you must fill the empty squares with numbers 1 to 8.



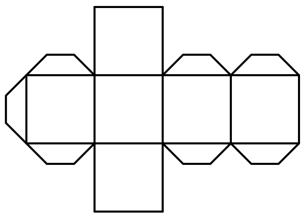
```
%FFEN_FillStyle 2
%FFEN_InnerStyle 1
%FFEN_InnerWidth 0.03
%FFEN_BorderAll 1
%FFEN 1'+1'=1/'+$1'-$1'+/1'-1'=1/'=$1'=$1'=/1'+1'="10
```

With the %FFEN_CornersRadius and the [,], ˜ and _ modifiers you can make "Dots and boxes" diagrams. What to play here to win?



```
%FFEN_SquareSize 1
%FFEN_InnerStyle 0
%FFEN_BorderWidth 0.1
%FFEN_CornersRadius 0.15
\scalebox{0.5}{ %FFEN 2[12/1[1_1[_1_1/1[_11[11/1[11[1[1/3[11
}
```

Finally you can use the { and } delimiters and directly insert psscript code to create whatever symbol which is not yet present. No space is allowed between { and }, but you can create TEX macros to put inside as in the following example.



```
\def\S{\psline[linewidth=0.03](0,0)(0.3,0.3)(0.7,0.3)(1,0)}
%FFEN_CornersRadius 0
%FFEN_CornersRadius 0
%FFEN_BorderWidth 0.03
%FFEN_InnerStyle 1
%FFEN #1#{\S}1#{\S}#{\S}/#*1{\S}4/#1#*2{\S}1#*2{\S}#*2{\S}
```

Some more remarks:

- FFEN_ThinLineWidth, FFEN_BoldLineWidth, FFEN_InnerWidth and FFEN_BorderWidth and everything related to them are relative to FFEN_SquareSize, so by default the real width for FFEN_ThinLineWidth is 0.03×0.5=0.015 cm.
- The width of Bars and Diagonals is the same as FFEN_BorderWidth.
- The centered dots (FFEN_FillStyle 6) have diameter of 3 times FFEN_BorderWidth.
- FFEN_BoldLineWidth is only used for Bold Cross (x) and Bold Circle (.).
- When FFEN_Fill is set to 2 (even), the upper left corner is empty, when it's set to 1 (odd) the upper left corner is filled.
- Symbols, figures and letters are inserted using "\rput[B](0.5*FFEN_SquareSize,0.25*FFEN_SquareSize){42}", so they look best with a FFEN_SquareSize of 0.5, size being changed afterward with a \scalebox.

ffen2tex v0.9 - Quick %FFEN reference

Alain Brobecker, august 2015

Draw a diagram: %FFEN UpperRow/.../LowerRow

Board Modifiers: *(default values are given, not reinitialised between two diagrams)*

%FFEN_SquareSize 0.5	
%FFEN_ThinLineWidth 0.03	
%FFEN_BoldLineWidth 0.15	
%FFEN_Fill 2	*(0=none / 1=odd / 2=even (Chess) / 3=full)*
%FFEN_FillStyle 0	*(0=lightgray / 1=gray / 2=black / 3=vlines / 4=hlines / 5=crosshatch / 6=centered dot)*
%FFEN_InnerStyle 0	*(0=none / 1=solid / 2=dashed / 3=dotted)*
%FFEN_InnerWidth 0.03	
%FFEN_BorderStyle 1	*(0=none / 1=solid / 2=dashed / 3=dotted)*
%FFEN_BorderWidth 0.06	
%FFEN_BorderUp 1	*(0=off / 1=on)*
%FFEN_BorderLeft 1	*(0=off / 1=on)*
%FFEN_BorderDown 1	*(0=off / 1=on)*
%FFEN_BorderRight 1	*(0=off / 1=on)*
%FFEN_BorderAll 1	*(0=off / 1=on, affects all 4 borders variables)*
%FFEN_PieceCount 0	*(0=off / 1=W+B / 2=?+?=W+B)*
%FFEN_CornersRadius 0	*(0=off / ?=radius)*

I also added the %FFEN_HBorderStyle, %FFEN_VBorderStyle, %FFEN_HBorderWidth, %FFEN_VBorderWidth, %FFEN_HInnerStyle, %FFEN_VInnerStyle, %FFEN_HInnerWidth and %FFEN_VInnerWidth modifiers.

Warnings: If you change more than once a value before using %FFEN only the first one is taken in account. The values are not reinitialised after a diagram. %FFEN_BorderAll is evaluated before the other %FFEN_Border modifiers.

Pieces:

char	piece	output	char	piece	output	char	piece	output
K / k	W / B King	♔ ♚	&	Heart	♡	u	Up Bar	╵
Q / q	W/B Queen	♕ ♛	”	Diamond	♢	,	Small Vertical Bar	ǀ
R / r	W / B Rook	♖ ♜	!	Spade	♠	|	Vertical Bar	│
B / b	W / B Bishop	♗ ♝	?	Club	♣	v	Up Left Bar	┘
N / n	W / B kNight	♘ ♞	A	W Up Left Arrow	⇖	y	Vertical Left Bar	┤
P / p	W / B Pawn	♙ ♟	a	W Up Arrow	⇧	+	Cross Bars	┼
C /c	W / B Circle	○ ●	.	Bold Circle	◯	U	Half Diagonal	⟍
D / d	W / B Dame	◎ ◉				\	Diagonal	╲
O / o	W / B Small Circle	∘ •				V	V Half Diagonals	∨
T / t	W / B Triangle	△ ▲				Y	Y Diagonal	⊠
S / s	W / B Square	□ ■				%	Cross Diagonals	╳
X / x	W / B Cross	⌧ ✖	:	Thin Cross	×	<	Less Than	<
E / e	W / B Star	✡ ✶	’a	Symbol	a	”42	Double Symbol	42
L / l	V / H Line Fill	▧ ▨	F	CrossHatch Fill	▩	f	White Fill	□

Valid symbols are !”#$%&()*+,-./0..9:;<=>?@A..Z[\]∧_‘a..z

Double symbols are inserted using two ’ characters, not the double quote!

TeX instructions and macros can be inserted as a piece using the { and } delimiters (no space!).

Piece Modifiers:

>	piece is between this square and right square
∧	piece is between this square and up square
∗N	rotated N*90 degree anticlockwise with N ∈ [0;3]
- or =	lightgray fill or gray fill (only for white pieces, ♡ and ♢ are not white)
@	piece is circled (same size as White Circle, but transparent)

Square Modifiers: *(can apply on many empty squares)*

$	change square(s) fill property
#	square(s) outside board (coloring affected by $ but not by %FFEN_Fill)
[and/or]	border on the left and/or right of next square(s)
˜ and/or _	border above and/or below next square(s)

ffen2tex v0.9 - Quick %CUBES reference and examples

Alain Brobecker, august 2015

Cubes Modifiers: *(default values are given, not reinitialised between two diagrams)*

%CUBES_WAngle 30	(between [0;90])
%CUBES_DAngle 30	(between [0;90])
%CUBES_WSize 1.0	
%CUBES_DSize 1.0	
%CUBES_HSize 1.0	
%CUBES_Rotation 0	(in [0;3] rotation around the vertical axis)
%CUBES_UpsideDown 0	(1=mirror around the vertical axis, made after rotation)
%CUBES_BoxBorder 000000000011	(bit 0,1=down, 2,3=backleft, 4,5=backright, 6,7=up, 8,9=frontright, 10,11=frontleft)
%CUBES_BoxInner 000000000011	(idem, those are binary numbers, bit 0 on the right, no need to put the leftmost zeroes)
%CUBES_BoxBorderStyle 1	(0=none / 1=solid / 2=dashed / 3=dotted)
%CUBES_BoxInnerStyle 1	(0=none / 1=solid / 2=dashed / 3=dotted)
%CUBES_BoxBorderWidth 0.06	
%CUBES_BoxInnerWidth 0.03	
%CUBES_CubesWidth 0.03	

Draw cubes:

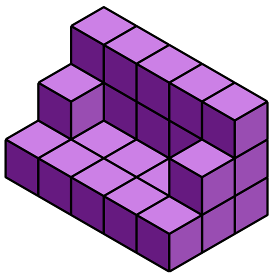
%CUBES W/D/H/.../.../.../.../.../.../.../.../...

.	empty
0-9	plain yellow/red/blue/black/white/green/orange/gray/cyan/magenta cube
a-z	for cubes with triple of colors defined beforehand in the file with:

```
\definecolor{a0}{rgb}{.99, .99, .99} %left
\definecolor{a1}{rgb}{.33, .33, .33} %right
\definecolor{a2}{rgb}{.66, .66, .66} %up
```

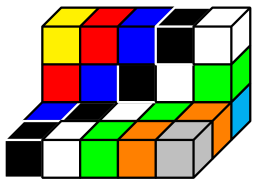
Examples:

Here is a sofa with Width=5, Depth=3 and Height=3. It uses a user-defined color.



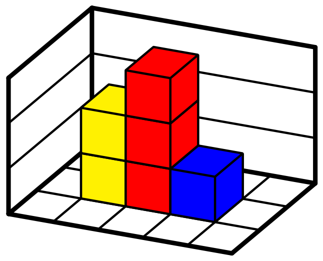
```
\definecolor{a0}{rgb}{.4, .1, .5}
\definecolor{a1}{rgb}{.6, .3, .7}
\definecolor{a2}{rgb}{.8, .5, .9}
%CUBES_BoxBorder 0
%CUBES_WSize 0.5
%CUBES_DSize 0.5
%CUBES_HSize 0.5
%CUBES 5/3/3/aaaaa/...../...../aaaaa/a...a/...../aaaaa/aaaaa/aaaaa
```

Almost the same sofa from another angle, and with predefined colors.



```
%CUBES_WAngle 0
%CUBES_DAngle 45
%CUBES_WSize 0.5
%CUBES_DSize 0.35355339 % divided by sqrt(2)
%CUBES_HSize 0.5
%CUBES 5/3/3/01234/...../...../12345/...../...../45678/23456/34567
```

The BoxBorder and BoxInner modifiers allow to have grids or lines around the figure:



```
%CUBES_WAngle 10
%CUBES_DAngle 40
%CUBES_WSize 0.6
%CUBES_DSize 0.48
%CUBES_HSize 0.6
%CUBES_BoxBorder 111111
%CUBES_BoxInner 101011
%CUBES 5/3/3/...../..1../...../...../.01../...../...../.012./.....
```