FEN2SVG handbook

Michaël George

July 14, 2019

Contents

Chapter 1

Introduction

1.1 What is this piece of software intented for?

It receives a FEN string or a file of FEN strings and transform it to chess diagram(s), in SVG.

1.2 How do I use it?

1.3 What does FEN means?

1.4 What does SVG means?

1.5 How does it work?

There is no real check of FEN legality. Row separators ("/") are just ignored. If less than 64 squares are found, the chessboard is filled with empty squares. If more than 64 squares are found, the surplus is simply ignored.

The main benefit of doing so is that the user has less constraints. The board can thus be empty or completely full. There can be any number of pieces (up to 64), even of the same kind, no king and so on.

How to convert to PNG?

1.6 Tools used

SVG editor: Inkscape and Geany,

SVG cleaner: SVGO,

chessmen: Wikimedia Commons(`https://commons.wikimedia.org/wiki/File:Chess_Pieces_Sprite.svg`),

C editor: Geany,

C compilator: GCC (Linux), MinGW (Windows),

C debugger: Valgrind, GDB,

C validator: Splint (`https://splint.org/`),

documentation editor: Texmaker,

absolute to relative paths: SVG transformations by Peter Collingridge (`http://petercollingridge.appspot.com/svg_transforms/`)

https://github.com/Klowner/inkscape-applytransforms

Chapter 2

SVG template

2.1 Why SVG?

- The end user (you) knows exactly how items are draw.
- The end user (you) can modify the output drawing.
- The diagram is scalable, whatever the resolution, even if this resolution increases within several years.
- You can customize a template of your own (wood squares, another move indicator, . . .).

2.2 How are the pieces aligned on their square?

Because queen is the tallest piece, it is used as reference. Queen is centered on its square Other pieces align their bottom with the queen bottom.

2.3 Why not SVGZ?

SVGZ are compressed SVG files. Therefore, to create SVGZ, two steps are required:

1. create the SVG (as it is currently the case),
2. compress it.

To compress, a C program needs a library. However, the goal is to keep FEN2SVG as simple as possible.
The good news is that you still can compress it by yourself, if needed.

2.4 How is the base template structured?

FEN2SVG uses a file where all the items are defined: template.svg.
There are two main parts:

- the <defs></defs>, were are defined
  - pieces,
  - coordinates,
  - squares,
  - border,
  - move indicator.
- the below part, where those defined items are used.

2.5 What is the square size?

Squares are 72 x 72.

2.6 Which font is used in the base template (for coordinates)?

Ume Gothic L

2.7 Why did you convert your coordinates font to path?

To avoid missing font issues.

2.8 Can I customize my own template?

Yes, of course, you can.

Do absolutly not modify the first ("<svg") and last line ("</svg>"), respect the definitions structure.

If you think it is worth it, share the result of your work.

Chapter 3

Source code

3.1 Why has C language been used for development?

C language is fast, lightweight and portable, moreover no graphical user interface is needed. The main drawback is precisly the lack of an interface.

3.2 How to compile for Linux?

gcc fen2svg.c unsortedlinkedlist.c -o fen2svg

3.3 Detecting memory leaks under Linux

Follow these two steps, in that order:

1. gcc -g -o0 unsortedlinkedlist.c fen2svg.c -o fen2svg
2. valgrind -v --leak-check=full ./fen2svg

GDB (GNU Debugger) could also be an useful debugger: `https://www.gnu.org/software/gdb/`.

3.4 How to validate code under Linux

splint unsortedlinkedlist.c fen2svg.c

3.5 How to compile for Windows under Linux?

32 bit: i686-w64-mingw32-gcc fen2svg.c unsortedlinkedlist.c -o fen2svg.exe

64 bit: x86_64-w64-mingw32-gcc fen2svg.c unsortedlinkedlist.c -o fen2svg.exe

3.6 How to compile for Windows?

Use your favourite compiler/IDE. If you have none, Mingw-w64 could be worth a try: `https://mingw-w64.org/`. Adapt the instructions in the above section.

Chapter 4

Possible improvements

- allow to check every FEN string (strict mode),
- allow to change square colour,
- change square texture (e.g. wood),
- add arrows, circles, squares and the like,
- limit output to a given area of the board,
- mark squares with dots, cross, . . . ,
- translucid squares,
- change move indicator,
- allow inner coordinates (rather than outside the board),
- allow fairy chess,
- use gradients for pieces or chessboard,
- add a caption at diagram bottom.

Chapter 5

Contact

Feel free to contact me at fen2svg.to.stronghold@spamgourmet.com, for improvements, corrections or sharing SVG chessboard templates.