The `fretplot` package
Version 0.0.3

Author and maintainer: Soumendra Ganguly
soumendraganguly.com

Copyright 2025– Soumendra Ganguly

Contents

1 Introduction

The *fretplot* LuaTeX package provides batch generation of scale and chord diagrams for plucked string instruments, such as the guitar.

It is designed for high flexibility, automation, and customization via simple yet powerful custom file formats for describing fretboard diagrams, which allow for easy batch generation of diagrams. To complement the high customizability, the package also comes with sensible yet attractive default settings for the diagrams.

The package also provides some easy-to-use LaTeX macros that understand the underlying music theory, allowing one to render, for example, a guitar scale diagram by specifying the musical scale or scale type via built-in macros, or even more directly via degree, pitch class, or interval formulae.

2 LaTeX macros

This package defines the following user-facing macros:

- \fptotikz

- \fptemplate

- \fpstemplate

- \fpscale

2.1 \fptotikz

Syntax: \fptotikz[OUTPUT_TIKZ_FILE]{INPUT_FP_FILE}

Description: compile an input fretplot file to Ti*k*Z.

Arguments:

- INPUT_FP_FILE: path to input fretplot file.
- OUTPUT_TIKZ_FILE (optional): path to output file to which generated Ti*k*Z will be written. If OUTPUT_TIKZ_FILE is not specified, then the output Ti*k*Z code will be included inline at the location of the \fptotikz invocation.

2.2 \fptemplate

Syntax: \fptemplate{OUTPUT_FP_FILE}

Description: generate a template fretplot file.

Arguments: OUTPUT_FP_FILE: path to output fretplot file.

2.3 \fpstemplate

Syntax: \fpstemplate{OUTPUT_FPS_FILE}

Description: generate a template fretplot scale style file.

Arguments: OUTPUT_FPS_FILE: path to output fretplot scale style file.

2.4 \fpscale

Syntax: \fpscale{ARGS}

Description: render diagram of (mode of) scale based on input scale formula (degree, pitch class, or interval) or generate corresponding fretplot file or Ti*k*Z code.

Arguments: ARGS: | and/or |\n-delimited list (possibly empty) of named argument assignments of the form key=value (with no surrounding spaces) in any order. Some of the subsequent pages of this section contain 2 tables:

   – A complete table of keys for `ARGS` and their respective default and possible values. If a key does not have a default value, then the corresponding entry in the table is empty.

        ∗ In the table, a *pitch class* refers to any element of the set

$$\{C, C\#, D, Eb, E, F, F\#, G, Ab, A, Bb, B\}.$$

        ∗ In the table, a *degree* is any element of the set

$$\{1, b2, 2, b3, 3, 4, b5, 5, b6, 6, b7, 7\}.$$

        ∗ In the table, an *interval* is a positive integer representing the number of semitones in a musical interval.

   – A table containing a list of macros that expand to interval formulas for popular scale types.

2.5 Examples

Create a directory titled `proj`. Create subdirectories `proj/include` and `proj/tikz`. Create a file `proj/macro-demo.tex`. So your project directory structure is as follows:

```
proj/
|-- include/
|-- tikz/
|-- macro-demo.tex
```

Copy the following code into `proj/macro-demo.tex`:

```latex
\documentclass[12pt,letterpaper]{article}
\usepackage{fretplot}
\usepackage{float} % For [H] placement specifier in figures.

\begin{document}

\begin{figure}[H]
\fpscale{parentroot=C#|formulatype=d|formula=1 2 b3 4 5 b6 7}
\caption{C$\sharp$ harmonic minor scale}
\end{figure}

% Create a fretplot file include/sclinclude.fp
% and a fretplot scale style file include/styles.fps.
% Comment these 2 lines out after the first run if
% you do not want to overwrite them in subsequent runs.
\fptemplate{include/sclinclude.fp}
\fpstemplate{include/styles.fps}

% Use fptotikz to test include/sclinclude.fp
% generated by \fptemplate.
\begin{figure}[H]
\fptotikz{include/sclinclude.fp}
\caption{Test \texttt{include/sclinclude.fp} generated by \texttt{\textbackslash fptemplate}}
\end{figure}

% Also test the TikZ file generation capability of
% \fptotikz, although the rest of the code does not
% depend on this. Note that this will merely create
% the file tikz/sclinclude.tex containing the
% TikZ code, but not include it in the document.
\fptotikz[tikz/sclinclude.tex]{include/sclinclude.fp}

\begin{figure}[H]
\fpscale{parentroot=A|formulatype=i|formula=\fpmaj|mode=2|
tuning=E B G D A E|numfrets=12|styletype=d|
labeltype=p|scalestylefile=include/styles.fps|
includefpfile=include/sclinclude.fp}
\caption{2\textsuperscript{nd} mode of A major scale (B Dorian)}
```

```
39  \end{figure}
40
41  \end{document}
```

Run the following command in `proj/` to compile `proj/macro-demo.tex`:

```
lualatex --shell-escape macro-demo.tex
```

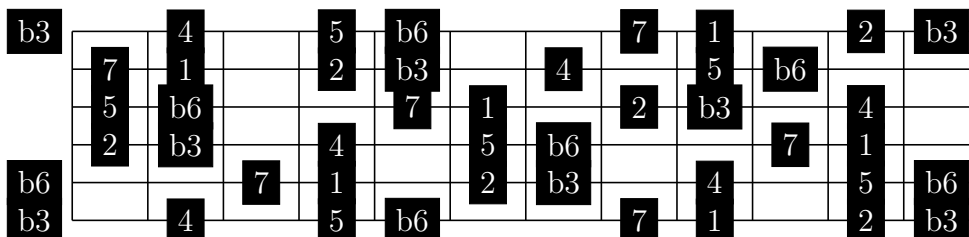You should get the following output in the compiled PDF file `proj/macro-demo.pdf`:
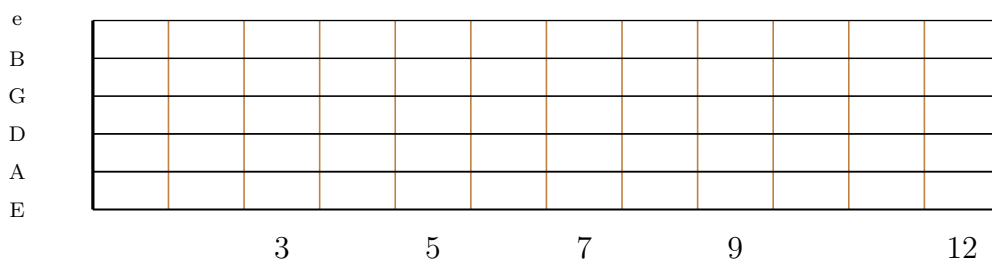

Figure 1: C♯ harmonic minor scale

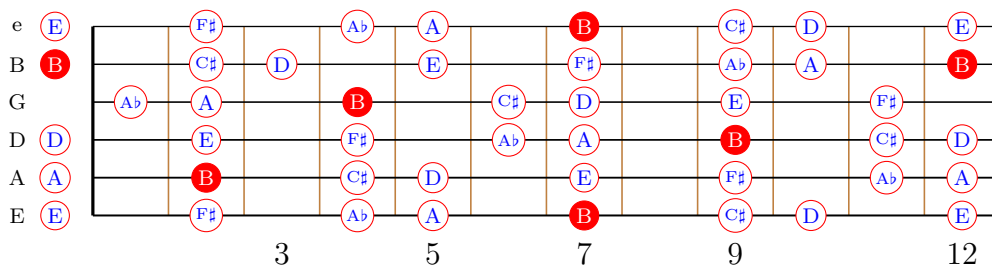Figure 2: Test `include/sclinclude.fp` generated by `\fptemplate`

Figure 3: 2$^{\text{nd}}$ mode of A major scale (B Dorian)

Play around with the generated files `proj/include/sclinclude.fp` and `proj/include/styles.fps` to customize the B Dorian figure. You can also edit the generated Ti*k*Z code `proj/tikz/sclinclude.tex` directly.

2.6 Tables

KEY	DEFAULT VALUE	VALUE
parentroot	C	Root of parent scale. This can be any pitch class. This is ignored if formulatype=p or if moderoot is specified. Please see the description of mode for more details.
formulatype	d	Type of the formula specified in the key named formula. Possible values are {d, p, i}. Here d stands for *degree*, p stands for *pitch class*, and i stands for *interval*.
formula	1 2 3 4 5 6 7	Formula of parent scale. This should be a list of degrees or pitch classes or intervals delimited by single ordinary space characters (ASCII 32). The type of this formula is indicated by specifying formulatype.
mode	1	Mode index of scale. This is specified as a positive integer. This mode will be rendered by \fpscale. Examples: if formulatype=p, formula=C D E F G A B, and mode=3, then the 3rd mode of the C major scale (E Phrygian) will be rendered. If formulatype=d, formula=1 2 3 4 5 6 7, mode=2, parentroot=A, and moderoot is not specified, then the 2nd mode of the A major scale (which is B Dorian) will be rendered. In the above scenario, if moderoot=A, then A Dorian will be rendered instead.
moderoot		Optional root of mode of scale. This can be any pitch class. This is ignored if formulatype=p. Please see the description of mode for more details.
tuning	E B G D A E	Tuning of strings of the instrument. This is specified as a list of pitch classes delimited by single ordinary space characters (ASCII 32).
numfrets	12	Number of frets on the fingerboard. If the instrument does not have frets, then one can style them to be transparent in the render.
styletype	d	Specify if the styling of notes should be degree-indexed or pitch class-indexed. Possible values are {d, p}. Here d stands for *degree* and p stands for *pitch class*. The user can customize the actual styles by specifying the key scalestylefile.
labeltype	d	Specify if the labelling of notes should be degree-indexed or pitch class-indexed. Possible values are {d, p}. Here d stands for *degree* and p stands for *pitch class*. The user can customize the actual labels by specifying the key scalestylefile.
outfpfile		Optional path to output fretplot file describing (mode of) scale. If this is specified, then no TikZ code will be generated by \fpscale (neither inline nor to outtikzfile). One can then make modifications to the output fretplot file and then generate TikZ code (inline or external file) from the fretplot file using \fptotikz. The keys includefpfile and scalestylefile are used to further customize the rendering. In particular, includefpfile is included at the end of outfpfile. Even if outfpfile is not specified, \fpscale still internally creates a temporary fretplot file which is then converted to TikZ. This temporary fretplot file also includes includefpfile at its end, which allows one to customize their render via includefpfile without specifying outfpfile. The same fretplot file can serve as the includefpfile for multiple \fpscale invocations, which allows for batch manipulation of scale plots. Temporary fretplot files created by \fpscale internally are deleted later automatically.
outtikzfile		Optional path to output file containing TikZ code describing (mode of) scale. This will be ignored if outfpfile is specified. If outfpfile is not specified and outtikzfile is specified, then \fpscale will not generate inline TikZ code. One can then make modifications to the output TikZ code and use \input to use it in their document.
includefpfile		Optional path to input fretplot file to be included at the end of outfpfile or an equivalent temporary fretplot file created by \fpscale for internal use. If there does not exist a file at the path specified by includefpfile, then a new empty file will be created at that path; one can then edit this new file for use in subsequent compilations. For more details, please see outfpfile.
scalestylefile		Optional path to input fretplot scale style file. This can be used to customize degree and/or pitch class-based styling and labelling for scale notes. Please see styletype and labeltype for additional details.

Table 1: Key-value pairs for \fpscale

MACRO	INTERVAL FORMULA IN SEMITONES	DESCRIPTION
\fpmpent	3 2 2 3 2	Minor pentatonic scale
\fpblues	3 2 1 1 3 2	Blues scale
\fpmaj	2 2 1 2 2 2 1	Major scale
\fpjmin	2 1 2 2 2 2 1	Jazz minor scale (ascending melodic minor scale)
\fphmin	2 1 2 2 1 3 1	Harmonic minor scale
\fphmaj	2 2 1 2 1 3 1	Harmonic major scale
\fpdhmaj	1 3 1 2 1 3 1	Double harmonic major scale
\fpnmaj	1 2 2 2 2 2 1	Neapolitan major scale
\fpnmin	1 2 2 2 1 3 1	Neapolitan minor scale
\fphunmaj	3 1 2 1 2 1 2	Hungarian major scale
\fpbmaj	2 2 1 2 1 1 2 1	Bebop major scale
\fpbmin	2 1 2 2 2 1 1 1	Bebop minor scale
\fpbdom	2 2 1 2 2 1 1 1	Bebop dominant scale
\fpldbeb	2 2 2 1 2 1 1 1	Lydian dominant bebop scale
\fppdbeb	1 3 1 2 1 2 1 1	Phrygian dominant bebop scale
\fpchr	1 1 1 1 1 1 1 1 1 1 1 1	Chromatic scale
\fpwt	2 2 2 2 2 2	Whole tone scale
\fptwowt	4 4 4	Two whole tones scale
\fpthreewt	6 6	Three whole tones scale
\fpmon	12	Monotonic scale
\fpaug	3 1 3 1 3 1	Augmented scale
\fptrit	1 3 2 1 3 2	Tritone scale
\fpwhdim	2 1 2 1 2 1 2 1	Whole half diminished scale
\fpmajsevenarp	4 3 4 1	Major seventh arpeggio
\fpdomsevenarp	4 3 3 2	Dominant seventh arpeggio
\fpminsevenarp	3 4 3 2	Minor seventh arpeggio
\fphdimsevenarp	3 3 4 2	Half diminished seventh arpeggio
\fpdimsevenarp	3 3 3 3	Diminished seventh arpeggio
\fpminmajsevenarp	3 4 4 1	Minor major seventh arpeggio

Table 2: Macros for interval formulae of popular scale types

3 The fretplot file format

A *fretplot file*, *fp file*, or `.fp` file describes the layout and appearance of a fretboard diagram.

3.1 Syntax

- Comment lines start with `#`. Comment lines and empty lines are ignored.

- Every other line of a fretplot file will have lines of form `firstword lots of data`, where `firstword` and `lots of data` must be separated by a single ordinary space character (ASCII 32), `lots of data` cannot be empty, and the first character of `lots of data` cannot be a whitespace character (space, tab, newline, or anything that matches `%s` in Lua).

-
  - Note notation: `<STRINGINDEX,FRETINDEX>`
    Example: `<6,5>`

  - Barre notation: `<STARTSTRINGINDEX-ENDSTRINGINDEX,FRETINDEX>`
    Example: `<1-6,5>`

- Barres never have labels. Frets, strings, and notes by default have no labels, but can be specified as follows:

  - Frets: `flFRETINDEX LABEL`
    Example: `fl3 3`

  - Strings: `slSTRINGINDEX LABEL`
    Example: `sl1 {\Large e}`

  - Notes: `nl<STRINGINDEX,FRETINDEX> LABEL`
    Example: `nl<6,5> {\Large A}`

- To include another fretplot file at the current location of a fretplot file, use `include FILEPATH`.

3.2 Parameters

All available parameters that can be used in a fretplot file are listed in the table on the next page. Operations are case-sensitive and must be written exactly as shown.

TYPE	POSSIBLE VALUES	OPERATIONS	OP DESCRIPTION
Boolean	True, False	flip PARAMETER	negate Boolean parameter named PARAMETER
Numerical	Any numerical value	+PARAMETER VALUE	add literal numerical value VALUE to numerical parameter named PARAMETER
		-PARAMETER VALUE	subtract literal numerical value VALUE from numerical parameter named PARAMETER
		*PARAMETER VALUE	multiply numerical parameter named PARAMETER by literal numerical value VALUE
		/PARAMETER VALUE	divide numerical parameter named PARAMETER by literal numerical value VALUE
		%PARAMETER VALUE	mod numerical parameter named PARAMETER by literal numerical value VALUE
		^PARAMETER VALUE	raise numerical parameter named PARAMETER to the power of literal numerical value VALUE
List	List of items delimited by single ordinary space characters (ASCII 32), or the *empty list*	void PARAMETER	empty the list parameter named PARAMETER
		>PARAMETER VALUE	extend list parameter named PARAMETER by literal list value VALUE
		<PARAMETER VALUE	remove values in literal list value VALUE from list parameter named PARAMETER
Style	List of PGF/TikZ styles delimited by single , (comma) characters	&PARAMETER VALUE	extend style parameter named PARAMETER by literal comma-delimited collection of styles VALUE

PARAMETERS	PARAM DESC	DEFAULT PARAM VALUE
onf0	draw notes directly on top of fret 0 (nut of guitar) if True and not if False	False
sovf	draw strings over frets if True and frets over strings if False	True
zoom	Factor by which the diagram will be scaled	1.0
rotn	Angle (in degrees) by which the diagram will be rotated	0
numfrt	Number of frets	12
numstr	Number of strings	6
frets	List of fret indices	0 1 2 3 4 5 6 7 8 9 10 11 12
strings	List of string indices	1 2 3 4 5 6
notes	List of notes	empty list
barres	List of barres	empty list
fx	Fret style	solid,line width=0.6,color=black
sx	String style	solid,line width=0.6,color=black
bx	Barre style	fill=black,draw=black
nx	Note style	shape=rectangle,draw=black,text=white,fill=black

Table 3: Parameters for fretplot files

3.3 Batch processing using operations on parameters

The operations are particularly useful for batch manipulation of diagrams.

For example, say, you are writing a book containing 100 chord diagrams. Each of those diagrams is described using its own fretplot file. Now you want to scale/enlarge each of them to 1.5 times of their respective original sizes, and additionally you want to rotate all of them by 30 degrees.

To do this, in each of those 100 fretplot files you can include another common fretplot file, say, `common.fp` by writing `include common.fp` *at the end* of all of those 100 fretplot files, and then in `common.fp` you can write the following two lines:

```
1  +rotn 30
2  *zoom 1.5
```

This will rotate all those 100 diagrams by 30 degrees and zoom in by a factor of 1.5, without you having to edit each of those 100 files individually.

Please note that lines in a fretplot file are interpreted sequentially, so the location where you write `include common.fp` in a fretplot file matters due to non-commutativity of certain operations. For instance, if the parameter `rotn` was `10` and you wanted to multiply it by `2` and subtract `5` from it, the order in which you performed those operations would matter.

The following is an example fretplot file that demonstrates the use of some parameters and some operations on them:

```
1   # Set a Boolean parameter.
2   onf0 False
3   # Negate a Boolean parameter.
4   flip onf0
5
6   # Perform some arithmetic operations on numerical parameters.
7   +rotn 90
8   /rotn 2
9   -rotn 45
10  %numstr 4
11
12  # Setting styles of some frets.
13  fx0 solid,line width=1.2,color=black
14  fx1 solid,line width=0.6,color=brown
15  # Setting labels of some frets.
16  fl3 3
17  fl5 5
18
19  # Setting styles of some strings.
20  sx1 solid,line width=0.5,color=black
21  # Setting labels of some strings.
22  sl1 {\scriptsize e}
23
24  # List of notes to be drawn.
25  notes <6,5> <3,6> <4,7> <5,7> <2,5> <1,5>
26  # Style a note.
27  nx<6,5> shape=circle,draw=red,text=white,fill=red,inner sep=1.5
28  # Label a note.
29  nl<6,5> {\scriptsize A}
30  # Change style of the note.
31  nx<6,5> shape=rectangle,draw=red,text=blue,fill=white,fill opacity=0.5,inner sep=1.0
32  # Relabel the note.
33  nl<6,5> 1
```

```
34  # Add some notes.
35  >notes <1,0> <1,1>
36  # Remove some notes.
37  <notes <5,5> <1,5> <2,2>
38  # Add more to existing style of a note.
39  &nx<2,5> draw=red,shape=circle
```

3.4   Example: barre chord diagram

- Save source below as `proj/include/amaj/src.fp`:

```
1   # Set diagram rotation angle.
2   rotn -90
3   # Set this to True to decrease space
4   # between fret 0 and the string labels.
5   onf0 True
6
7   # We do not want all frets for this chord diagram.
8   numfrt 3
9   frets 0 1 2 3
10
11  # Label the frets appropriately.
12  fl1 5
13  fl3 7
14  # Label the strings.
15  sl1 {\small e}
16  sl2 {\small B}
17  sl3 {\small G}
18  sl4 {\small D}
19  sl5 {\small A}
20  sl6 {\small E}
21
22  # A major triad barre chord.
23  ## Specify the barre.
24  barres <1-6,1>
25  bx<1-6,1> fill=black, draw=black
26  ## Specify the notes.
27  notes <6,1> <3,2> <4,3> <5,3> <2,1> <1,1>
28  ## Set note styles and lables.
29  nx<6,1> shape=circle,draw=red,text=white,fill=red,inner sep=1.5
30  nl<6,1> {\scriptsize A}
31  ##
32  nx<5,3> shape=circle,draw=red,text=blue,fill=white,inner sep=1.7
33  nl<5,3> {\scriptsize E}
34  ##
35  nx<4,3> shape=circle,draw=red,text=white,fill=red,inner sep=1.5
36  nl<4,3> {\scriptsize A}
37  ##
38  nx<3,2> shape=circle,draw=red,text=blue,fill=white,inner sep=0.3
39  nl<3,2> {\scriptsize C$\sharp$}
40  ##
41  nx<2,1> shape=circle,draw=red,text=blue,fill=white,inner sep=1.7
42  nl<2,1> {\scriptsize E}
43  ##
44  nx<1,1> shape=circle,draw=red,text=white,fill=red,inner sep=1.5
45  nl<1,1> {\scriptsize A}
```

- Save source below as `proj/amaj.tex`:

```
1   \documentclass[12pt,letterpaper]{article}
2   \usepackage{fretplot}
3   \usepackage{float} % For [H] placement specifier in figures.
4
5   \begin{document}
6
7   \begin{figure}[H]
8   \centering
9     \fptotikz{include/amaj/src.fp}
10    \caption{A major triad barre chord}
11  \end{figure}
12
13  \end{document}
```

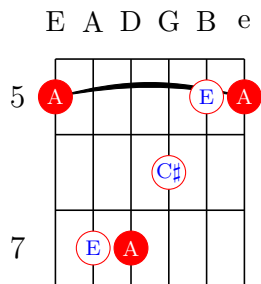- Run `lualatex --shell-escape amaj.tex` in `proj/` to get the following output:

Figure 4: A major triad barre chord

3.5   Example: fretless guitar diagram

- Save source below as `proj/include/fretless/src.fp`:

```
1   # Set diagram zoom level.
2   zoom 0.8
3
4   # Fretless guitar.
5   frets 0 12
6
7   # TIP: uncomment the following line to remove the nut and the bridge.
8   # void frets
9
10  # Add some notes.
11  notes <6,5> <3,6>
12  ## Set note styles and lables.
13  ##
14  nx<6,5> shape=circle,draw=red,text=white,fill=red,inner sep=1.5
15  nl<6,5> {\scriptsize A}
16  ##
17  nx<3,6> shape=circle,draw=red,text=blue,fill=white,inner sep=0.3
18  nl<3,6> {\scriptsize C$\sharp$}
```

- Save source below as `proj/fretless.tex`:

```
1   \documentclass[12pt,letterpaper]{article}
2   \usepackage{fretplot}
3   \usepackage{float} % For [H] placement specifier in figures.
4
5   \begin{document}
6
7   \begin{figure}[H]
8   \centering
9     \fptotikz{include/fretless/src.fp}
10    \caption{Some notes indicated on a fretless guitar}
11  \end{figure}
12
13  \end{document}
```

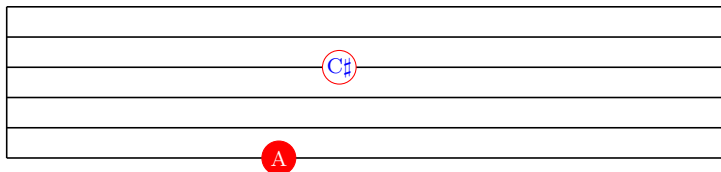- Run `lualatex --shell-escape fretless.tex` in `proj/` to get the following output:

Figure 5: Some notes indicated on a fretless guitar

3.6   Example: custom instrument diagram

- Save source below as `proj/include/custom-instrument/src.fp`:

```
1   # Set diagram zoom level.
2   zoom 0.5
3   # Set diagram rotation angle.
4   rotn 185
5
6   # Set fret and string counts.
7   numfrt 5
8   numstr 8
9
10  # Set the frets.
11  # Fret 0 is the nut.
12  frets 0 1 3 4 5
13
14  # Set strings.
15  strings 1 2 3 4 5 6 7 8
16  # Remove a string.
17  <strings 3
18
19  # Label frets.
20  fl1 {\tiny 1}
21  fl4 {\tiny 4}
22  fl5 {\tiny 5}
23
24  # Style the frets.
25  fx1 dashed,line width=0.6,color=black
26  fx3 solid,line width=0.6,color=red
27
28  # Label strings.
29  sl1 {\tiny 1}
30  sl2 {\tiny 2}
31  sl4 {\tiny 4}
32  sl5 {\tiny 5}
33  sl6 {\tiny 6}
34  sl7 {\tiny 7}
35  sl8 {\tiny 8}
36
37  # Draw the frets over the strings.
38  sovf False
39
40  # Add a note.
41  notes <6,0>
42  nx<6,0> shape=circle,draw=red,text=white,fill=red,inner sep=0.2
43  nl<6,0> {\tiny A}
44  # I want this note to be drawn on fret 0 (nut).
45  onf0 True
```

- Save source below as `proj/custom-instrument.tex`:

```
1   \documentclass[12pt,letterpaper]{article}
2   \usepackage{fretplot}
3   \usepackage{float} % For [H] placement specifier in figures.
4
5   \begin{document}
6
7   \begin{figure}[H]
8   \centering
9     \fptotikz{include/custom-instrument/src.fp}
10    \caption{Instrument with 8 strings and 5 frets with a fret and a string missing}
11  \end{figure}
12
13  \end{document}
```

- Run `lualatex --shell-escape custom-instrument.tex` in `proj/` to get the following output:
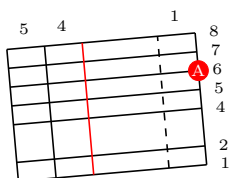

Figure 6: Instrument with 8 strings and 5 frets with a fret and a string missing

4 The fretplot scale style file format

A *fretplot scale style file*, *fps file*, or `.fps` file customizes the appearance and labels of notes in a scale, based on pitch class or degree.

4.1 Syntax

- A *pitch class* is any element of the set

$$\texttt{\{C, C\#, D, Eb, E, F, F\#, G, Ab, A, Bb, B\}}.$$

- A *degree* is any element of the set

$$\texttt{\{1, b2, 2, b3, 3, 4, b5, 5, b6, 6, b7, 7\}}.$$

- Each line assigns a style or label to a pitch class or degree:

    - Label based on pitch class: `lPITCHCLASS LABEL`.
    - Label based on degree: `lDEGREE LABEL`.
    - PGF/Ti*k*Z style based on pitch class: `xPITCHCLASS STYLE`.
    - PGF/Ti*k*Z style based on degree: `xDEGREE STYLE`.

4.2 An example

```
1   lC {\scriptsize C}
2   lC# {\tiny C$\sharp$}
3   lD {\scriptsize D}
4   lEb {\tiny E$\flat$}
5
6   xC shape=circle,draw=red,text=blue,fill=white,inner sep=1.0
7   xC# shape=circle,draw=red,text=blue,fill=white,inner sep=1.0
8
9   l1 {\scriptsize 1}
10  lb2 {\tiny $\flat$2}
11
12  x1 shape=circle,draw=red,text=white,fill=red,inner sep=1.0
13  xb2 shape=circle,draw=red,text=blue,fill=white,inner sep=1.0
```