FIM : Fbi IMproved

21.03.2013

A flexible *image viewer*

`muc.ccc.de`

svn export `http://code.autistici.org/svn/fim`

what ?

fim is an image viewer

- a modal one: two modes:
  - *interactive* (using $\boxed{\texttt{Keys}}$ *)*
  - *command line*
- scriptable:
  - *from inside* (with an inner scripting language)
  - *externally* (from the shell)
- multi-interfaced
  - SDL (Xorg,...)
  - imlib2 (Xorg,...)
  - Linux framebuffer device
  - aalib (ascii art rendering, from textual terminal)

why the name *Fbi IMproved* ?

because it originated as a fork of `Fbi` version 1.31 (by Gerd Hoffmann)

why ?

an inspiring quote (pertaining the `mutt` mail user agent):

"All mail clients suck. This one just sucks less." Michael Elkins, circa 1995

how to invoke ?

base cases, from the shell:

- `$ fim file.jpg` *opens a file*
- `$ fim file1.jpg -- --file2.jpg` *opens two files*
- `$ fim /media/pictures/` *opens all the files in the directory*
- `$ fim --help` *provides a help message*
- `$ man fim` *manual page for the 'fim' command*
- `$ man fimrc` *manual page for the fim internal language*

how to use it ?

basics of interactive use:

- $\boxed{\texttt{n}}$ *go to the next file*
- $\boxed{\texttt{p}}$ *go to the previous file*
- $\boxed{\texttt{+}}$ *magnify*
- $\boxed{\texttt{-}}$ *shrink*
- $\boxed{\texttt{q}}$ *quit*

how to use it ?

do I know these keys from elsewhere ?

- $\boxed{\texttt{h}}$ *pans to the left*
- $\boxed{\texttt{j}}$ *pans down*
- $\boxed{\texttt{k}}$ *pans up*
- $\boxed{\texttt{l}}$ *pans to the right*

how to script it ?

the command line (or, *readline*):

- `:` *enters in readline mode*
- `:` `Enter` *quits the readline mode*
- `:cmd` *invokes command "cmd"*
- `:help` *invokes inline documentation*
- `:next` *go to the next file*
- `:prev` *go to the previous file*

alternative invocation modes ?

fancy standard input use:

- `$ find /media -iname *.tiff | fim -` *opens a files list*
- `$ vgrabbj -d /dev/video0 | fim -i` *opens an ‘anonymous’ file from the pipe*
- `$ cat cmds.fim | fim -p` *executes commands by reading them from stdin*

alternative invocation modes ?

fancy standard input use:

- `$ fim * > selection.txt` *saves a list of selected filenames in a files*
- `$ cat selection.txt | fim - > top-ten.txt` *refines a files selection*

why to script ?

because it may be useful:

- `:help "goto"` *requests “goto” help*
- `:goto "2"` *jump to the second file*
- `:goto "+2"` *go forward two files*
- `:goto "-2"` *step two files backwards*
- `:goto "^"` *go to the first file*
- `:goto "$"` *go to the last file*
- `:goto "33%"` *go to the file positioned around one third in the files list*
- `:goto "/.*flower*.jpg/"` *jump to the first file matching this regular expression*
- `:goto "+//"` *jump to the next file matching the last inserted regular expression*

ok but scripting is boring!

there are shortcuts:

- `:2` *go to the second file*
- `:^` *go to the first file*
- `:$` *go to the last file file*
- `:/.*flower*.jpg` *go to the first file matching this regular expression*

even shortest shortcuts ?

keyboard sequences valid in interactive mode:

- `2n` *forward two times*
- `3p` *backward three times*
- `4+` *magnify four times*
- `.` *repeat last action*

immage scaling ?

examples

- `:scale "20%"` *scale to 20%*
- `:scale "+20%"` *magnify by one fifth*
- `:scale "-20%"` *shrink by one fifth*
- `:scale "0.2"` *shrink to one fifth*
- `:scale "w"` *scale to width*
- `:scale "h"` *scale to height*
- `:scale "a"` *scale automatically*

keys ?

binding commands to keys:

- `:bind 'n' 'next;'` $\boxed{n}$ will execute `next`
- `:bind 'N' '10next;'`
- `:bind 'C-n' 'goto "+//"'` *example of key use:* Ctrl

autocommands ?

execution of certain commands in particular conditions

- `:autocmd 'PostGoto' '' 'reload;'` *reload na image after a jump.*
- `:autocmd 'PostInteractiveCommand' '' 'redisplay;'` *show an image after an interactive command*
- `:autocmd 'PostGoto' '.*mirrored.*' 'mirror;'` *mirror images whose name matche the mirrored pattern*

language features

- `:alias 'ne' 'next'` *alias assignment*
- `:os=scale;` *variable assignment*

various constructs

- `:echo i:width` *certain fim variables are accessible*
- `:if(i:width>_screen_width){scale 'w'}` *conditional execution*
- `:while(i:width>_screen_width){goto '+1'}` *iterating a cycle*

configuration file

a configuration file is basically a fim program read at startup

- `$HOME/.fimrc`
- `/etc/fimrc`

integration with other software

- with mailcap+mutt
- with midinight commander (`mc`)
- multi-page files rendering
- in GNOME scripts

what does fim contain ?

- about 30000 C++ lines (out of the 2000 original Fbi's lines)
- cleaned-up Fbi code, but still dirty
- contains a parser written with `GNU flex/bison`
- uses the `GNU readline` library (the same used by `bash`)
- employs C++'s STL library

why and when to use it ?

- ergonomic reasons
- when we don't have X (minimalistic or embedded systems)
- when Fbi does not collaborate with `screen`
- particular needs

experimental features

- **PDF** decoding
- **DJVU** decoding
- read at a specified offset
- ...

how to install it ?

on debian/ubuntu, with `apt-get`:
`sudo apt-get install fim` # beware: loooong time not updated!!!

how to install it ?

on debian/ubuntu, from source:

```
sudo apt-get install subversion

sudo apt-get install automake autoconf libtool libreadline6-dev

sudo apt-get install libjpeg-dev libpng-dev libtiff-dev libgif-dev

sudo apt-get install libsdl-dev libaa1-dev

sudo apt-get install libpoppler-dev libdjvulibre-dev libspectre-dev

svn co http://code.autistici.org/svn/fim/trunk/ fim

cd fim

./configure --enable-poppler --enable-aa --enable-sdl

make
```

questions ?

- website:
  `http://savannah.nongnu.org/projects/fbi-improved`
- SVN repo: `http://code.autistici.org/svn/fim/trunk`
- `$ help!`

the end

happy hacking!