

nu-mode

Modern keybinding for Emacs

Pierre-Yves Luyten()

This manual is for nu-mode (version 0.8.2 of 2017-12-13), a modern keybinding for Emacs.
Copyright © 2017 Pierre-Yves Luyten.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Table of Contents

1	What is Emacs by the way	1
2	Overview	2
2.1	Navigate	2
2.2	Selections and Deletion	3
2.3	Cut Copy Paste	4
2.4	Undo or Redo	4
2.5	Dependencies	4
2.6	Help	5
2.7	Menus and Prompters	5
2.8	Advanced Edition	7
3	Files Buffers Windows	8
3.1	Internal Windows and Tabs	8
3.2	Files, Bookmarks	8
3.3	External Windows	8
4	Completion	9
5	Hacking	10
6	Advanced Usage	11
7	Flavours	12
8	Setup	13
8.1	I prefer vi keys for basic movement	13
8.2	I don't like the Alt modifier for the paddle	13
8.3	Prompts	13
8.4	Menus	13
8.5	Main keymap & free keys	13
8.6	Completion	14
9	Integration	15
9.1	Major-mode : dired	15
9.2	Major-mode / Minor-mode : org	15
9.3	Major-mode : magit	15
9.4	Major-mode : term	15
9.5	Major-mode : texinfo	16

9.6	Minor-mode : auto-complete.....	16
9.7	Minor-mode : god-mode.....	16
9.8	Minor-mode : evil-mode.....	16
9.9	Other : autojump.....	16
10	Drawbacks & Limits.....	17
11	Contributing.....	18
	Appendix A GNU Free Documentation License ..	19

1 What is Emacs by the way

Emacs is not a text editor, its an interface, because it allows you to control, from one program, others, in order to read mail or run commands, manage files and so on.

Actually Emacs is a choice between several interfaces. Which one you need, it all depends on your usage, and your tastes.

VIM is more popular than Emacs. This is for a reason : its keys are more efficient. Learning VIM is easier than learning Emacs.

This documentation does cover nu-mode and nu-state, but you should also consider evil, spacemacs, and ergoemacs.

evil is an emulation of vim. it makes emacs behave like vim. spacemacs does like evil, and adds tons of stuff, sometimes at a cost.

ergoemacs is a different approach : it is an ergonomic keybinding for emacs. Not modal. It hopes to makes non modal editing possible, efficient. So it does rebinds most common emacs commands in order to make keybindings fine enough.

nu-state is based on evil. the normal state is like vim, and by default each buffer opens in normal state. so, like vim. visual mode is also the same. but * like spacemacs, nu offers a space key to open menus. Except these menus follow operator-motion logic. nu is minimal. * insert state is not vim insert state. it is a full keymap, and actually, it is perfectly usable on its own.

nu-mode is different than nu-state. It is close to ergoemacs. It prefers easy to remember shortcuts (control+c, control+v...) and sort-of menus, which are actually quite powerful since this principle allows to provide quick and easy access to many commands. nu-mode is a mode, and its keymap.

If you use dvorak or bÃ©po, ErgoEmacs might be more adapted, or nu-state.

2 Overview

WARNING : this package has not yet reached 1.0. It is subject to change.

“nu-mode “ is a modern keybinding for Emacs. It aims at respecting modern conventions like *Ctrl+c*, *Ctrl+v*, while making editing as comfortable as possible. It also aims at making Emacs pretty easy & discoverable. It is, currently, primarily designed for azerty and qwerty. Dvoark or BÃ©po users might prefer ErgoEmacs at the moment.

Technically, nu-mode is a global minor mode, its keymap (nu-keymap), and few commands which are not provided in raw Emacs. Using hooks, nu-mode does integrate with vanilla Emacs features like isearch or org-mode, or external features like undo-tree or ivy completion mechanism.

Modern, means respecting conventions like

1. *Control-f* to find
2. *Control-r* to replace
3. *Control-s* to save
4. *Control-x*, *Control-c*, *Control-v* to cut, copy, paste.

Actually there are several flavours of nu-mode, so *Ctrl+c* for example, will not have the same meaning depending on the flavour you load. Each flavour is adapted to one usecase. These are described in their own part in this documentation.

To allow user to leverage large panel of functions, nu-mode relies on (its own) menus. For example, *Control+f* will raise a “find-prompt” offering several search related features. A prompt is a pop-up displaying a menu. Each feature is accessible with a key. This prompt allows to discover features, and run many different commands easily.

There are, on the other hands, “immediate keys”. For example, to isearch-forward-regexp (that is, “Find”), *Alt+f* will directly run the command. So, it’s still possible to trigger find using one direct shortcut, but another one will allow to discover many features : grep, occur, and so on.

The prompt will display a list of keys - actually, sequences - and associated functions : for example *t* to find-tag, *w* to ace-jump-word-mode, *m* for imenu, *b* for regexp-builder, *o* for occur, *Alt+f* to isearch-forward-regexp.

Prompts are responsive according to mode or text being currently selected : according to major mode or selection state, different functions will be available or not.

2.1 Navigate

I
J K L

nu-mode navigation is based on a paddle : while arrows still work, nu-mode relies on *Alt* key to navigate. By default the paddle is an inversed T : *i j k l* go up, left, down, right (previous line, backward char, next line, forward char). If *Shift* is pressed while moving, this will select or extend selection, as *Shift* plus arrow does.

If user prefers to use vi like keys to navigate, it is possible to set (`setq nu-use-vi-paddle`) This variable will maintain **Alt** to navigate, but **h j k l** will have the same meaning as in vi.

The classic Nu paddle is the same as ErgoEmacs. There are several reasons leading to this choice ; the most important being **Alt** an easy key to press with the thumb, thus sparing pinky finger. Another obvious reason is that many CUA keys are mainly on the bottom left part of the keyboard, while **Control** key associated with the paddle, has a specific meaning.

(Actually, any nu-mode function, is available from the **Alt** key modifier, since prompts offer access to functions.)

Additionally, use **Alt u** and **Alt o** to move to previous, next word.

U I O
J K L

Use **Alt+\$** to go to end of line, and **Alt+h** (or, **Alt+^** to go to beginning of line. **Shift** only works with **h** - but you can use mark prompt for this.

To reach a line, or the beginning of buffer, or the end of the buffer, you could use goto prompt, available from **Control+g**. Just read this goto-prompt, which offers to navigate per line, paragraph, sentence and so on. To find a char, a string, a line starting with a specific letter, use avy-jump : also inside **Alt+f** prompt. To directly trigger a regexp-search, use **Control+f**, which is another way to navigate.

Numeric arguments can apply to navigation. Press **Alt+1** then **Alt+5** to input 15 as a numeric argument ; now input **Alt+k** to go down one line : this will go down 15 lines.

‘Repeat’ also applies to navigation. Enter **Control+Return** to repeat a navigation (or any command).

However - let’s finish with navigation. Use **Alt+Space** to scroll (to the bottom), and **Alt+Backspace** to go back to the top.

Note that Emacs sometimes require you to scroll another window. This is quite useful to go through **Help** buffer, for example. If you need this, use **Control+Alt+Space**. To scroll the other way around the other window, use first a negative argument (**Alt+-**).

This navigation chapter is meant as a general introduction to emacs-nu ; because navigation is a basis, but also because we saw many emacs-nu principles :

1. Direct keys for most common operations.
2. Prompts for slightly less common to rare operations.
3. Numeric arguments.
4. Repeat.
5. Few emacs convention respected, but some.

These same principles apply to other prompts.

2.2 Selections and Deletion

Now that you can input text plus navigate, let’s examine several alternatives to delete (cut) text. When there is some important text to work with, it is possible to use **Alt+a** to set

mark, then move to extend selection. A little help will popup so you can use advanced features.

Rectangle selection need to invoke the selection prompt : **Control+a**. Similarly to classic selection, rectangle selection offer a little popup to help the user discovering how to work.

nu-mode uses evil-mode to implement the keybinding. Generally vim keys are not invited to the party, but selection actually authorize vi keys. So for example it's possible while selecting text, to **3w ~** to toggle capitalization up to next 3 words. This specific selection state allows to leverage vim power for selection. (Who needs to input text while selecting?)

Sometimes it is more convenient not to select text, but rather to directly delete some text.

1. Use **Alt+x Alt+x** to cut the current line.
2. Use **Alt+x** + a vim motion to delete like vim. For example, **Alt+x e** to delete up to end of word. Numbers are authorized.
3. Backspace / Delete are still available.
4. Use **Control + u** to delete previous word.
5. Use **Control + \$** to delete up to end of line.
6. Use **Control +x** to trigger a delete prompt. This will offer you to delete what you want (function, org-node, sentence, ...).
7. As seen above, alternatively, first mark (select) text you want.

2.3 Cut Copy Paste

Use **Alt + c** to copy current line, or, while a selection is active, copy this selection.

Use **Alt + v** to paste the current clipboard. Or, you can use **Control + v** to invoke an advanced "Paste" prompt.

2.4 Undo or Redo

As expected, use **Alt+z** to undo or **Control+z** to access the full features or undo visualize.

This screen displays last changes, and you can navigate these using the paddle : **i** (redo), **j** (switch to left branch), **Alt+k** (redo), **l** (switch to right branch). Press **q** to quit this screen or **Alt+q** to abort. You can also toggle selection mode with **s** or toggle timestamps using **t**.

2.5 Dependencies

nu-mode depends on several libraries, both native or external ones.

1. recentf
2. undo-tree

These are all optional dependencies. First user might one of following completion framework.

1. helm
2. ivy

Then user might want to be able to access some features.

1. help-fns+, to describe-keymap

2. ace-jump
3. magit

2.6 Help

Use **Control+q** to quit a sequence or a command in progress. Use **Control+z** to undo last command. Use **Control+h** to gain access to help functions. This includes the usual shortcuts to describe what a key does, or what a function does. Use **Alt+q** prompt to quit Emacs (or run some other function).

Prompts are usually triggered using a **Alt** modifier and advertise a function using the same key, but with **Control** modifier. For example, **Alt+s** will pop-up a save prompt offering several features, while **Control+s** directly saves the current buffer.

These prompts offer you to quickly learn where functions are.

2.7 Menus and Prompters

This section should explain

1. the menu that is, the content : the menus are the group of functions that can be accessed. menus are not organized as menus (toolbar) in vanilla Emacs. These are more close to Spacemacs : menus are grouped by themes. Major mode should include its func in various menus, not in one single menu.
2. the prompters are the method to trigger a func inside the menu, prompters are the interface.

If you want to see all available menus, press **Control+h** **<space>** to invoke cheat sheet, then look for prompts functions.

The menus are organized by actions, like : finding, opening, deleting, replacing. The menus are **not** organized by objects, like : buffers, frames, registers. Regarding this organization, the disadvantage is to split a set of functions accross various menus. Advantages - It feels intuitive to use a logic like **<new> <window>**, **<delete> <window>** rather than **<window> <new>**, **<window> <delete>** though it might depend on native language of user - It helps having a balanced number of menus, with balanced items. - It is more consistent with CUA shortcuts. CUA shortcuts like **control+f** find, **control+n** new, are not representing objects, but functions.

1. help menu is not a prompt but is a map. Press **Control+h** to view many help function. Install **help-fns+** to obtain even more functions.
2. nu-a-prompt (selection) this menu is an alternative to set mark. it provides few functions regarding selection.
it also offers to select a directory.
3. replace This group includes replace, transpose, sort functions. it does not include markdown, indentation => see bold prompt
4. print component allows printing, but also evaluating, compiling, converting. it is also macro evaluation.
5. quit menu allows few quit related func.
6. save menu offers to save file, register, bookmarks, session, link... anything you want to keep for later!

7. find menu allows to search, occur, find tag, grep, build regexp...
8. goto menu is for other navigation (not found in other prompts). This includes going to other window (ace-window).
you might also look at window, and obviously find prompts.
9. window menu is for window management.
This includes scrolling, sizing, transpose. This does not include window creation (**Alt+t** or new menu)
10. delete menu prompt allows to delete text, file, buffer, window.
11. copy menu menu is currently limited.
12. insert menu allows to insert text, snippets, command, abbrev... If you want to evaluate, see Print.
13. bold menu is for capitalize, indent, markdown it does not offer to transpose, sort. See Replace.
14. new menu is either for creation (file, dir, mail) or invoke a process (term, calc, command)
15. open should open something already existing : file, register, buffer, bookmarks, mails

The standard prompt is the “light” prompt. It does only popup a little message. A key can directly be pressed to trigger the associated function. If the user does not know yet the function, or the key, he might use

1. **q** does always quit prompt.
2. Pressing **<tab>** will raise a “full-prompt”. First, a full prompt will display available keys (what key sequences do not have). Second, a prompt will also advertise direct shortcuts to commands : ie, shortcuts that would have directly run some function without going through the prompts. For example, using **Control+g** prompt, it is possible to press **g** to ‘ace-window’, but prompt will also advertise that **Alt+g** would have directly called this same function without requiring to go through a prompt. This does not mean using the prompt was wrong. It is rather meant as a way to discover and as a reminder, since nu-mode has many alternatives.
3. Pressing **<space>** will raise a “completion-prompt”. This will use a completion mechanism to select the desired function.
4. Pressing **?** will allow to call describe-function on a function, rather than running it
5. Prompts allow you to use arguments in two ways. The first way, is to use standard Emacs keys to trigger an argument (eg, **Control+1**, or **Alt+1**, or **Alt+-**, or use the universal argument which is bound in nu-mode to **Alt+p**), then run the prompt, then choose the command that will be affected by the argument. The second way to use arguments is to first run the prompt, then directly press a digit or - to increment the argument. Then, press the relevant key to run the desired function.
6. A prompt can be toggled to some ‘repeat’ state. Run a prompt, then press **+**. You can then use the prompt as usual, except that after running the function, prompt will ask you again for a key, with the same keymap. Escape the prompt pressing any unbound key.

The “do-prompt” from **Alt+d** is different. “do-prompt” directly displays all possible keys. It offers several frequent features.

While it would be possible to handle buffers or open files using prompts, “do-prompt” offers direct keys for this, making it possible to avoid other prompts most of the time. On the other hand the “do-prompt” does not add anything to Nu. It is only a matter of convenience, but usage has shown this to be very useful.

keys *menu* or, \hat{A}^2 , trigger “menu map”. Menu map then o does run the open menu, menu map then n the new menus, and so on. This makes possible to completely skip the **Control** key.

2.8 Advanced Edition

Some advanced editon features rely on **Alt+r**, which will invoke replace prompt, allowing you to replace-regexp, merge-lines, delete spaces, or invoke other functions (**Control+r** directly triggers replace-regexp).

1. **Alt+y** will copy to current line char under point (on the below line).
2. **Alt+e** will copy to current line char above point (on the above line).

Control+s prompt allows you to save current column as a goal-column. While you move next and previous lines, cursor will try to reach this column as far as possible.

Alt+b will run nu-bold, which will act differently according to major-mode. This try to emphasis, which might have a different meaning given the mode. **Control+b** will run an emphasis (bold) prompt, allowing to indent, fill-paragraph, and so on.

3 Files Buffers Windows

3.1 Internal Windows and Tabs

Use *Control+o* prompt to open a buffer. *Alt+n* allows you to create another internal window ; *Control+n* would prompt, allowing for example to vertical split or open another external frame or so. *Alt+w* allows you to kill buffer.

Use *Alt+t* to open another tab (horizontal-split). *Control+t* will prompt for which buffer to open in another window.

The delete prompt (*Control+x*) allows to delete either this window or all other windows. The save prompt (*Control+s*) allows to save current configuration, which you can open from open prompt (*Control+o*).

To navigate current internal windows, use the tab prompt (*Control+t*). Note that, in order to switch to another tab, you could also call *ace-jump-char-mode* (*Control+f 1*). If you actually just need to scroll another tab, stay in current one and use *Control+Alt+Space* to 'scroll-other-window'.

3.2 Files, Bookmarks

Use *Alt+s* to save current file, *Control+s* to invoke a prompt to rename it. *Control+o* to open a file, or *Alt+d i*. You can also have bookmarks : use the same open-prompt and save-prompt for this.

You can open a directory, too.

direx is the emacs file manager. You can easily use *regexp* to open, delete, copy, rename files. And since you are using *nu-mode*, this is *regexp-file-management-for-human-beings* ,). See the relevant section on Integration.

3.3 External Windows

It is possible, from Emacs to handle a bit of window management. But you should rather use a decent window manager. Look however at the save prompt...

4 Completion

It is possible not to use any completion system using emacs-nu. However i recommend to use **ivy**.

```
(nu-setup-ivy)
```

An alternative is **helm-mode**

```
(nu-setup-helm)
```

These convenience functions will bind some features, like find-files, M-x, search, to corresponding function in Ivy or Helm. Ivy completion system displays some help. Helm does not popup help yet.

5 Hacking

The print buffer, invoked from **Control+p**, will offer you to eval things or make (compile). The insert prompt, invoked from **Control+v**, will allow you to insert a file, or the result of an async shell command into a new buffer. The save prompt, from **Control+s**, offers to use git power thanks to magit.

Control+p also offers to grep, find-grep or ediff.

As seen above, **Control+b** will run an emphasis-prompt, which allows to comment. Onto c-mode or various lisp-mode, **Alt+b** will toggle comments.

Control+s prompt also allows you to create tags to find definitions.

6 Advanced Usage

1. To trigger a major mode sequence, start with *Control+Space*. This will invoke vanilla emacs *Control+c*. After this shortcut is pressed, a list of available functions will appear on the bottom of the screen.

If you already defined you own keys starting with *Control+c*, do not amend this definition.

```
(define-key mykeymap kbd("\C-c h") 'myfunction)
```

To invoke above myfunction example, press *Control-SPC h*.

2. You should not rely on *Control+x* in its vanilla Emacs meaning, however to trigger x prefix, use *Alt+d e*. This will raise a prompt to trigger Control-X-Prefix.

‘Repeat’ is invoked from *Control+Return*. It was hacked to work with prompts : thus, you can repeat a command ran from a prompt. Prefix arguments (numeric-argument, negative-argument) can work : either use standard *Alt+1*, *Alt+2* ... shortcuts plus one command. Or, from a prompts, directly type a figure (or, a number) then choose the command (eg, from delete prompt, type 3 then invoke kill-word to kill 3 words ; or type 2 then 3 to input 23). Alternatively, first trigger the argument from standard shortcuts, then call the prompt. What is invoked from the prompt will make use of that argument.

Note that ‘where-is’ function, which sometimes advertises shortcuts, will not work as usual, since a prompt is not a keymap : ‘where-is’ do not know how to invoke *Control-r* then *r* to invoke replace-regexp. Thus for each function accessible from a prompt, if you read ‘You can run this command with <menu> r k’, please understand, despite it is also true, that you can also run the command with *Control-r-k*.

7 Flavours

The manual did expose standard keybindings, for default flavour. This use case is adapted for rather intensive usage of Emacs, with much typing. It might be not adapted in two different cases.

1. Very intensive usage. If Emacs is used, not several hours, but many hours, then one should rely on modal mode.
2. Not so intensive, but without much typing. If Emacs usage is more code review than typing text/code, then again modal mode editing be more adapted.
3. Usage mixed with other applications which are not keyboard-friendly Some applications rely on mouse, or, force shortcuts like **Control c**. In this case, it is better to get Emacs behave like the other applications. So, for example, one can **Control c** in one app and **Control v** in another, one of these being Emacs (no matter which).
4. Light usage of Emacs If Emacs is a limited application in time, then it is more difficult to determine which mode to use. Default mode or Control mode are adapted.

Control mode : menus are not accessed using Control+key, and immediate functions using Alt. It is the opposite. Control keys trigger immediate functions, so for example Control+c to copy, and menus are accessed using Alt+key, like Alt+c for copy menu. This was the first design when nu-mode was developped, because it makes using both Emacs and other applications easy, or because it is easy for fingers. It makes learning nu-mode very easy. It has a drawback however : using this Control-mode flavour for extended time, it becomes frustating to have to use a) either control keys too much or b) constantly the alt key then another key. In other words immediate keys are not easy enough, while menus are actually too easy to reach. To enable control mode, throw this in your dot emacs : after the (nu-mode 1), put (nu-set-control-mode).

8 Setup

As usual with Emacs, customization comes down at the very first usage. So even before to describe edition, customization has to be mentioned. There is not a single `defcustom` inside `nu-mode`. All customization is to be done in other ways.

8.1 I prefer vi keys for basic movement

Below code will make `h j k l` behave like `VI`. Beginning of line become `i`.

```
(setq nu-use-vi-paddle)
```

8.2 I don't like the Alt modifier for the paddle

While it is easy (simply bind `Control+ [ijkluo]` to the movement, this is not recommended, since `alt` key is suppose to offer **any function**.

8.3 Prompts

If you want prompts to automatically prompt a buffer rather than invoking `which-key`, use :

```
(defalias 'nu-prompt-for-keymap 'nu-buffer-prompt-for-keymap)
```

Or also

```
(defalias 'nu-prompt-for-keymap 'nu-completion-prompt-for-keymap)
```

Finally you might also write your own prompt. See for example `which-key` prompt for a simple example.

8.4 Menus

if you want to amend a menu, either to customize your own setup or to develop a package to integrate a mode with `nu-mode`, you should define a function which add keybindings to one or more `nu-mode` menus.

```
(defun my-nu-mode-menu-customization ()
  (if (eq nu-major-mode 'org-mode)
      (define-key nu-window-map "a" 'delete-window)))
```

Then add the function to `nu-populate-hook`

```
(add-hook 'nu-populate-hook 'my-nu-mode-menu-customization)
```

The hook documentations does provide some conventions.

8.5 Main keymap & free keys

If you want to amend `nu-mode` keymap, amend `evil-insert-state-map`. Or, you can define-key which are not used by `nu-mode`.

Free keys are

1. `Control+c` + somme letter. This is the standard Emacs convention.
2. `Control` or `Alt` + punctuation, except `$` or `^`, as in `vi`.
3. `Control+Alt` + anything

4. `Control+Shift` + anything
5. `Alt+Shift` + anything

8.6 Completion

Please refer to chapter Completion for ivy & helm. Nu-mode might later integrate more completion systems.

9 Integration

nu-mode is to be responsive - in order for major modes to preserve nu-mode philosophy, prompts plus some direct keys will change according to the context : mainly major-mode. More globally, nu-mode does integrate with other libraries.

9.1 Major-mode : dired

As expected, most of vanilla **Dired** keys are respected : these are keys which should, in Fundamental mode for example, run self-insert-key - for example, **d** will mark a file for deletion. You might also use **h** so available keys will be displayed at the bottom of the window.

Dired allows to use nu-mode shortcuts to navigate : the **Alt+i** & **k** paddle allow to navigate the list, while **j** will go up to parent directory and **l** will find-file at point (for example, visit directory at point).

Some prompts are adapted to dired : selections (with **Alt+a**) for example, will provide the different dired options to mark files. Look at “replace”, “insert”, “find” prompts and so on. Where it make sense the usual prompts will be useful inside dired. Keys that *remain* useful inside dired are kept.

Use **Control+p** (uniserval-argument) then **s** key to edit the ls command.

9.2 Major-mode / Minor-mode : org

nu-mode has many features related to org. Rather than storing all org-related features in one place, nu-mode does enrich its usual prompts or even commands given the org-mode context. You can also use the **Control+Space** to discover all org mode features.

Alt+n new prompt allows to capture a note. Using open-prompt, it is possible to open agenda. **Alt+s** prompt will allow you to org-store-link. **Alt+v** prompt will allow to paste an org link. This same prompt offers to org-table-insert-column or row. It is possible to insert a timestamp or directly a deadline. **Control+b** will run org-emphasis. **Alt+b** print prompt allows to expand (“print”), which does correspond to what org-modes binds to Alt+Tab, which is an *awful* choice. It is also possible to delete a node from the delete prompt. One can org-mark-ring-goto using goto prompt.

Note that shift selecteotions do not work with org-mode.

9.3 Major-mode : magit

Use **Control+s**, ie save-prompt, to gain access to magit-status.

Magit already is very easy to use, since you might press **h** to discover its features.

9.4 Major-mode : term

term is a major mode which has two components (sub-modes)

1. char mode
2. line mode

Default (startup) is char mode.

Char mode is like a real terminal. This allows you to do real stuff, including terminal embedded interfaces. On the other hand, shortcuts are not available since keys are directly sent to term. There is an exception however : press **Control+c 1** (or, **Control+space 1**) and you will access a unique prompt. This prompt allows you to navigate to other buffers, trigger line-mode, cancel current command, or call another prompt.

Basically in term mode you only run commands, plus navigate.

When you want to select output or navigate more seriously, just use the line sub-mode. In line-mode, nu-keymap is not amended. The advantage is to gain nu-keymap + some emulation, obviously on the other hand this is not anymore a full term emulation since keys are not directly sent.

From line mode, press control+space to gain a dedicated prompt, allowing several terminal related features like returning to char mode.

9.5 Major-mode : texinfo

Texinfo has little add-ons to prompts : “print” prompt will offer to makeinfo or convert to pdf, and “insert” prompt to texinfo-insert...

9.6 Minor-mode : auto-complete

There isn't much to say. Auto-complete does integrate well, that's all!

9.7 Minor-mode : god-mode

God mode works but, is not binded. You would have to assign a key yourself to trigger god-mode.

9.8 Minor-mode : evil-mode

While nu-mode is not a modal editor, and aims at being the most efficient keymap, sometimes using lot of modifiers might still be harassing. Technically using vim keymap is feasible : you can activate evil-mode, then switch from evil-state to emacs-state as you want.

Either make VI normal-state use the nu-mode paddle.

```
(define-key evil-normal-state-map (kbd "h") 'evil-insert)
(define-key evil-normal-state-map (kbd "j") 'evil-backward-char)
(define-key evil-normal-state-map (kbd "i") 'evil-previous-line)
(define-key evil-normal-state-map (kbd "k") 'evil-next-line)
```

Or, make nu-mode use the vi paddle.

However i would not recommend working like this. nu-mode is thought to work in non-modal, modern editing. It does not need modes.

Using vanilla evil or spacemacs might be an alternative.

9.9 Other : autojump

autojump is part of the keybinding (Control f prompt).

10 Drawbacks & Limits

1. There might be bugs on emacs-nu
2. Emacs-nu is meant for azerty, qwzerty. It would not be very good with other layouts. However the plan for other layouts might be simple. Control + key should stick to the mnemonic, like Ctrl+s to display save prompt. While immediate keys like Alt+s or, obviously, the paddle Alt+i / Alt+j / Alt+k / Alt+l, should be transposed according to layout. Control+h key to display Cheat Sheet, plus prompts, will help memorizing.

11 Contributing

Emacs nu is moving to a “one file per feature” style. There are files for common, core nu-mode, then files for integrating it. For example there is a nu-isearch.el file, a nu-ibuffer.el file. I do expect to isolate a file per mode.

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any,

- be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
 - C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their

titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.