nu-mode

Modern keybinding for Emacs

Pierre-Yves Luyten()

This manual is for nu-mode (version 0.8 of 2014-09-12), a modern keybinding for Emacs.
Copyright © 2014 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 Overview

“nu-mode “ is a modern keybinding for Emacs. It does offer a consistent CUA Emacs interface, and plans to integrate properly with most popular Emacs modes. CUA, here, does imply a deep meaning, despite it does partly use cua-mode.

nu-mode is a global minor mode, its keymap, and some commands which are not provided in raw Emacs.

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.

To allow user to leverage large panel of functions, nu-mode relies on prompts. For example, while `Control+f` will directly offer to isearch-forward-regexp (that is, “Find”), `Alt+f` will raise a “find-prompt” offering several search related features. The prompt will display a list of keys - actually, sequences - and associated functions : 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, and so on.

Prompts do adapt to context : ie, according to major mode or selection state, different functions will be available or not. As of October, 2014, there are 250 “define-key” inside the relevant code portion, which means about 15 prompts offer to call 25O commands. Obviously, additionnal keys paddle offers to easily navigate.

1.1 How long is nu-mode to learn

Basically, understanding nu-mode requires to 1. learn the paddle (two minutes should make it), 2. get the idea and memorize the prompts (fifteen minutes) then 3. get used to helm configuration (add fifteen more minutes).

If you’re really impatient, go to Cheat Sheet...

1.2 Dependencies

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

1. recentf
2. cua-selection-mode
3. helm
4. undo-tree

These are all optional dependencies.

1. help-fns+, to describe-keymap
2. ace-jump
3. magit

1.3 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.

1.4 Prompts

Prompts are basically a window showing several possible keys, and which functions are to be run given this input. However some features deserve some deeper explanation. When a key does "raise a prompt", minibuffer will ask for an input or to press ? to get more details. This is basically a key sequence with a minibuffer prompt (this function is 'nu-light-prompt-for-keymap and is usable with any keymap, like help-map, and this is what Control+h does, for example).

Pressing <tab> will raise a "buffer-prompt". First, a buffer 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 *Alt+d* prompt, it is possible to press o to 'kill-word, but prompt will advertise that *Control+d* 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.

Third, a prompt (no matter light- or buffer-) allows 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.

Fourth, a buffer-prompt also allows to get some help about functions. Run the buffer-prompt, press ?, then the same keymap as usual will not run functions but describe functions. For example, while 'replace-prompt' ususally run 'revert-buffer' on a press, once ? has been pressed, a will instead describe function 'revert-buffer'.

Fifth, a prompt (no matter light- or buffer-) 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. You might notice is does allow some sort of modal editing, but this is not really the intented behaviour. This feature is rather made to make some commands easier, for example deleting a paragraph then two word then a character. Modal editing is possible using nu-mode, see below.

Sixth, a buffer-prompt can always be scrolled using <space> and <backspace>. Thus space and backspace are not bound. I might be tempted however to inverse things later on, bind space/backspace, and have buffer prompt use M-space / M-backspace to scroll.

Seventh, a prompt (no matter light- or buffer-) can always be escaped using 'q'. This letter is never bound. You can always use it. Note that *Control+q* and *Alt+q* are neither bound.

1.5 helm-prompts

An alternative way to explore prompts is helm. Not the most efficient in my humble opinion, however it works : from a light prompt, instead of pressing `tab` to trigger a full (buffer) prompt, it is possible to press `space` to have a helm completion using the candidates corresponding to the prompt.

Prompts shortcuts are shown but totally ignored : this is helm which decides candidate selection. Prompts shortcuts are only shown for information, in order to learn - shortcuts might also be used to select candidate however).

Direct shortcuts are also shown, as usual.

1.6 ok but where are prompts

Either jump to Cheat Sheet part of this manual (or, on github : https://github.com/pyluyten/emacs-nu/wiki/Cheat-Sheet---All-Prompts). Or, run nu-describe-keymap (*Control+h Alt+k*) and precise nu-menu-map.

All of the below keys use the alt modifier to run the associated prompt (for example *Alt+f* to run the find prompt) except the h and o which uses Control modifier.

```
key             binding
---             -------
a               nu-a-map
b               nu-bold-map
d               nu-delete-map
f               nu-find-map
g               nu-goto-map
h               help-map
o               nu-open-map
p               nu-print-map
q               nu-quit-map
r               nu-replace-map
s               nu-save-map
v               nu-insert-map
w               nu-window-map
```

Prompts content are more or less predictible ; though some notes might help. 'nu-a-map' is the selection prompt ; usually Control+a stands for select all but nu-mode chooses a saner default. 'bold-map' is actually an 'emphasis' prompt, including stuff like 'toggle comments' in relevant modes. goto-map also allows to navigate internal windows. 'print' rather means 'eval'. 'replace' is rather about 'changing'.

From Emacs, the simplest is to `Control+h` then `*` to light-prompt about the different available prompts. This just tells which key to press, not exactly which shortcut, and allows to describe the associated keymap.

You can `Control+h * *` to describe the full nu-keymap.

2 View and Edit

Obviously just type keys to input text. With slight notes : *Alt+m* to carriage return + indent. Use *Alt+v c* to insert literally a character.

2.1 Navigate

```
  I
J K L
```

nu-mode navigation is based on a paddle, an inversed T : while arrows still work, nu-mode relies on Alt key to navigate. 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.

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. This is exactly the same principle than ErgoEmacs applies. 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, appart from opening, 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, use *Alt+g* then the appropriate key. 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 ace-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. Emacs vanilla keybindings for 'help-mode' and some other offer Space and Backspace to do this - this is one of the few conventions somewhat respected into Emacs-nu. Yup.

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 slighly 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.

1. Use *Control-x* to cut the current line.
2. Use *Control+j*,*Control+l* to delete backward, forward char. Backspace / Delete are still available.
3. Use *Control + u* to delete previous word.
4. Use *Control + $* to delete up to end of line.

1. Use *Alt+d* to trigger a prompt. This will offer you to delete what you want (function, org-node, sentence, ...).
2. Alternatively, first mark (select) text you want to select then use *Control+x*.

Right, but how to select? Once again, several alternatives...here we go

1. The paddle allow to directly select ("mark") text : keep *shift* pressed, then move either with arrows or *Alt+<some key of the paddle>*. Using *Alt-Shift-u*, for example, will select previous word (or extend current selection to previous word).
2. An alternative in order to select text is to press *Control+a* to mark current "block" (a word including punctuation), then move to extend selection.
3. An alternative is to press *Alt+a* to invoke "a-" prompt, allowing to select "a-" word, "a-" sentence, and so on, or to set the mark. Or, to set a rectangular mark. Read the prompt! Notably, *Alt+a i* will just set the mark, while *Alt+a j* will set the mark & go backward-char - and other paddle keys will perform some logical actions. Moreover, the a-prompt tries hard to keep in sync with goto prompt to share the same keys, which are quite closely related to paddle keys - thus it is easy to remember both mark and goto prompt...
4. You can also use a direct key to set a rectangular mark : *Control+Shift+a*.

Once some text is selected you can "toggle the point and mark" using either *Control+a* or *Alt+a*, or deactivate the selection using *Alt+q*, or cut the text using *Control+x* (or *Alt+d* might be nice depending on your fingers position).

2.3 Cut Copy Paste

When no selection is active, use *Control* + *x* to cut current line, or a deletion command since deletions actually cut text (as a reminder : emacs 'kill-' commands will copy to kill-ring, while emacs 'delete-' commands will not).

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

Use *Control* + *v* to paste the current clipboard. Following *Control* + *v* will replace this paste with precedent clipboard item. However, if you need to paste several times, first invoke a numeric prefix argument to specify how many times to paste, then type *Control+v*. Or, you can use *Alt* + *v* to invoke an advanced "Paste" prompt.

2.4 Undo or Redo

As expected, use *Control+z* to undo and *Control+Shift+z* to redo. Or, use *Alt+z* to invoke undo-tree visualizer to play with discard changes in an advanced way.

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.

Note that this is somewhat a prompt, despite not a nu-prompt, which is a reminder on why to use *Alt*.

2.5 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).

Alt+v prompt to open a line. *Alt+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 possibe.

Control+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. *Alt+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

Emacs has a specific word for its internal frames : it says 'Windows'. This is awful, but might be understood given we precise these are internal windows, not X or Wayland windows. Emacs has no support for standard tabs because there would be too much tabs. But hidden buffers are tabs, aren't they?

Use *Control+o* prompt to open a buffer. You can use *Control+Shift+o* to directly be prompted for a buffer to open. Or, *Control+Shift+i* to direclty open next buffer, skipping *Messages* or *Backtrace* and the like. *Control+n* allows you to create another internal window ; *Alt+n* would prompt, allowing for example to vertical split or open another external frame or so. *Control+w* allows you to kill buffer.

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

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

To navigate current internal windows, use the goto prompt (*Alt+g*) then one of the four "windmove" functions. Note that, in order to switch to another tab, you could also call ace-jump-char-mode (either *Alt+f f* or directly *Control+Shift+f* then specify which character of the other tab you want to jump to). 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 *Control+s* to save current file, *Alt+s* to invoke a prompt to rename it. *Control+o* to open a file. You can also have bookmarks : use the same open-prompt and save-prompt for this.

You can open a directory, too.

dired 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 proompt...

4 Minibuffer

It is highly advised to enable **helm-mode** while using emacs-nu. Use below code in your .emacs.d/init.el or whatever file you are using to customize Emacs :

```
(helm-mode 1)
```

Some prompts do provide access to helm functions : open prompt to access help-mini or help-find-files. Or insert prompt to access kill-ring from helm-show-kill-ring. Mx will use helm-Mx, ibuffer does use helm too, and so on.

Once helm buffer is opened to let you select file / buffer / function or whatever helm migth help you select, use paddle (*M-i* , *M-k* to navigate up/down). If there are "sources", use *Shift+Space* / *Shift+backspace* to navigate it or common *Alt+Space* / *Alt+Backspace* to scroll page. As expected, *Control+q* or *Alt+q* to quit helm.

While looking for a file, C-u will not delete backward word but will go up one directory. Any helm prompt will use `TAB` as a "persistent action", which sometimes means completion (while looking for files), while most of the times this will run the 'actions' while keeping helm session alive. For example, this would open the file but keep the helm prompt there to allow you opening another one.

Several items might be selected using *Control+Space* to mark them. Otherwise, item at point will be the unique one.

Default action will be appropriate most of the time. Once you have selected / marked the item(s) you want, you can change the action using *Control+Return* then either the *F[0-9]* key as indicated or *return* after having selected the right action.

5 Hacking

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

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

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

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

6 Advanced Usage

If you are not used to Emacs, reading this chapter is not necessary. But Emacs veteran will probably be interested. Emacs-nu redefines many keys, however two points make learning curve shorter

1. For any user, knowing CUA keybinds will make emacs-nu discoverable.
2. For veteran Emacs user, two important sequences remain:

M-x is still there - which means, any unmapped or hidden function can be easily triggered. It will run helm-M-x, and *tab* would raise a help buffer while keeping the session alive.

Control+h will invoke a help prompt (thus, user can get help about help...).

Now, two things should probably be kept in mind while trying nu-mode

1. To trigger a major mode sequence, start with *Control+Space*. This will invoke vanilla emacs *Control+c*. For example, into org-mode, use *Control+Space* *Control+n* to navigate to next node. 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*, or at least no regularly. However to trigger x prefix, use *Control+Shift+x*. This will raise a prompt to trigger Control-X-Prefix. Note this should *not* be forced to, otherwise this is a bug you're encouraged to report.

'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. (Did you try 'where-is git push' in magit? this does not work. What is acceptable for maggit is not for a full keymap.)

Thus :

1. For each function accessible from a prompt, an additional shortcut is created, accessible from *menu* plus the same *key* than the prompt, in order to make where-is advertise. Hence, 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*.
2. From the help prompt (*Control-h*), run `h` to invoke nu-help. This function will present you all prompts, and will offer you to describe their keymaps - that is, all the functions you can access from these prompts.

7 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.

7.1 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. Modifiers keys will mainly invoke nu-mode commands. This allows to combine both worlds : if you don't know how to run something, use a prompt and read it in order to learn how to directly invoke the key. Eg, using Delete prompt allows to dired-delete a file, but this same prompts also shows a - direct, native - key to run the same. This should make dired learning curve better.

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.

7.2 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.

`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.

The Control+Space Control+Space shortcut will trigger what would have been Control+c Control+c. Note that shift selectections do not work with org-mode.

7.3 Major-mode : magit

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

A work has started to integrate magit properly. For example, replace prompt will offer to toggle whether next commit has –amend. It is possible to use new prompt to create branch, open prompt to open branch manager, and so on.

7.4 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 (or, control+space) 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. (TODO : adapt it.) 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 prompt to come back to char mode (plus, many other stuff...)

7.5 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...

7.6 Minor-mode : auto-complete

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

7.7 Minor-mode : god-mode

god-mode is one possible modal editing. See below about evil-mode. god-mode can be toggled using *Control+g*. Using vanilla god-mode, it is then possible to quit god-mode pressing *Control+g* again (the modifier is necessary).

I'm currently waiting for a patch to be integrated, in order to allow god-mode to allow configuring which modifier is added to keys. This patch will allow to use god-mode with alt automatically triggered. Forward / Backward char / line / word will be one key distant, while delete prompts becomes nicer. Commands relying on *Control* modifier will be triggered with g key. Using this, quitting god-mode is *gg*, since first g will toggle Control modifier.

7.8 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. Using vim keymap is feasible : you can activate evil-mode, then switch from evil-state to emacs-state as you want.

Even while in evil-state, you will enjoy a few nu-mode keys, like Alt+v, Alt+f, and probably many others. However vim paddle (hjkl for left down up right) and nu-mode

paddle (ijkl for up left down right) disagree on three of these four keys! If you want to have evil with nu-mode paddle, I recommend you to make vim paddle similar to nu-mode. Since `i` will not be available anymore to insert, use `h` for this purpose. Simply put below lines on your .emacs:

```
(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)
```

Obviously this only fixes the basic paddle. Backward and Foward word are different, but fixing the paddle is fine.

7.9 Other : autojump

autojump is part of the keybinding (alt f prompt, or directly Control+Shift+f to autojump to a char).

8 Cheat Sheet

key	Alt	Control	S-Ctrl	S-Alt
a	*Mark* / toggle	Mark block / toggle	rectangle	
z	*Undo tree*	Undo		
e	Copy Above			
r	*Replace*	Replace regex		
t	*Other tab*	New tab		
y	Copy Below			
u	word back	ctrl+bkspc.		select
i	up	X [tab]		select
o	forward word	*Open*		select
p	*Print*	Argument		
q	*Quit*	Quit		
s	*Save*	Save		
d	*Delete*	kill-word		
f	*Find*	Find		
g	*Goto*	God		
h	orig.	*Help*		
j	left	backsp		select
k	down	kill line		select
l	right	kill right		select
m	newline & indent	X [RET]		
w	*Window* Pr	kill-buffer		
x	helm Mx	Cut	*ctl-x*	
c	-	Copy		
v	*Insert*	Paste		
B	*Emphasis*	"bold"		
n	*New*	split window		

9 Customization

As ususal 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.

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

Easy. Do bind Control+ [ijkluo] to the movement. Eventually bind Alt + the same to delete, or rely on alt+d. You loose some consistency, since alt key is suppose to offer **any function**. (On the other hand, you could remap Alt+o to open prompt.)

9.2 Prompts

If you want prompts to automatically prompt a buffer rather than just a 'nu-light-prompt-for-keymap, which is the default, use :

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

If you dislike both nu-buffer-prompt-for-keymap and default nu-light-prompt-for-keymap (why? please report a bug!!), you're free to develop your own function. The only requirement is this function to offer to call-interactively the functions being part of the prompt. Be warned, you'll have to handle a huge amounts of details however if you want your function to be pleasant =)

As of August, 2014, some prompts have a "fixed" map you can amend directly: - nu-window-map (Alt+w) - nu-new-map (Alt+n) - nu-a-map (Control+a)

If you want to amend some "dynamic prompt", add a hook after its "populating" - nu-populate-print does populate print-map (Alt+p) - nu-populate-delete (Alt+d, nu-delete-map) - nu-populate-bold-map (Alt+b, nu-bold-map) - nu-populate-insert-map (Alt+v, nu-insert-map) - nu-populate-save-map (Alt+s, nu-save-map) - nu-populate-open-map (Control+o, nu-open-map) - nu-populate-goto-map (Alt+g), nu-goto-map)

You can also raise prompt for maps outide of nu-mode : - ctl-x-map, using *Control+Shift+x* - help-map

Finally, it is totaly feasible to use nu-prompts styles for your own keymap : just (nu-prompt-for-keymap some-keymap), and you're there : one nu-mode prompt will offer you to leverage some-keymap.

9.3 Main keymap

If you want to amend nu-mode keymap, amend nu-keymap.

9.4 Side effects

nu-mode does trigger some settings which are not mandatory

cua-selection-mode is really logical given the nu-mode paddle. However, feel free to fix this call in nu-mode initialization.

```
(cua-selection-mode 1)
```

Also, cursor-type is amended. You can set it back, but again, despite not mandatory for nu-mode to work this is really logical given the keybinding.

```
(set-default 'cursor-type 'bar)
```

10 Drawbacks & Limits

Many keys are used for prompts, making it necessary for more functions to use a prompt (while still many functions are bound to direct shortcuts, thanks to Control+Shift or Alt+Shift). Anyway typing three keys to run a function appears to be fine enough. As of today there are 94 define-keys in nu-mode.el, including prompts (twice for each prompt). This still represents a bunch of shortcuts to learn...

A more serious complaint I could formulate is that describe-keymap or where-is cannot totally works with nu-mode. As a workaround, menu key is bounded to prompts (thus, menu o would run the open prompt), allowing where-is to partially works. But where-is cannot work with responsive prompts : it cannot guess prompt will offer different features according to different modes. Anyway I don't see any magic here, nor any serious trouble - this just means the user will have to vaguely look at the Integration section of this manual...

While starting, the worst part might be the two ambiguous keys : - o is a paddle key while navigating or selecting. But o is a prompt key for opening file. This is both ambiguous, plus this forces to use control for a prompt. On the other hand, some usage proves this to be rather easy. I'm all eager to get some feedback on this point. - h is a synonym for beginning of line, which makes it exactly the same as above : h will move, but control+h is a prompt.

An even more serious remark would be that, once someone has learned a way, he will not want to have to learn a second time. Learning Emacs, twice??! Well, given Emacs-nu needs 30 minutes to learn, I don't think this is that bad. Most of major-modes / minor-modes will not require additional time. Helm-mode might require a few minutes to get used to.

If Emacs-nu breaks some mode, this would be a more serious issue. In such case, please fill-in a bug.

11 Contributing

As of today nu-mode only has one author, & contribution is more than welcomed. Please look at github.

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. Secondarily, 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.