CUA Mode

Operation	Keystroke	Function	Note
Emacs CUA Mode	One of the main purpose of the cua-mode is to provide the main keys of the Common User Interface (CUA) that have become popular over the years in many editors and tools: • `C-c` : copy • `C-v` : paste • `C-u` : cut • `C-z` : undo These bindings, however, conflict with standard Emacs bindings: • `C-c` and `C-x` are prefixes to many other keys, • `C-d` and `C-z` are used for other commands. • The CUA mode `C-c` and `C-u` are only available when a region is marked. • If you want to access the Emacs commands that use the `C-c` or `C-x` prefix when a region is marked you can type the prefix key twice very fast or type the entire chord very fast (less than .2 seconds). • The cua-mode package provides another very useful feature: the cua-rectangle-mark-mode which allows operating on all lines of a rectangle area simultaneously, a great feature not already available in Emacs. • The cua-mode is heavily customizable. The custom variable cua-enable-cua-keys can be set to nil to disable the use of cya-mode version of copy, paste, cut and undo. It is also possible to activate the cua-rectangle-mark-mode without activating the cua-mode itself. • The PEL package uses the standard Emacs keys for copy, paste and cut, with some enhancements (see the Cut & Paste) table. And for the undo it uses the undo-tree package. The PEL package provides access to the cua-rectangle-mark-mode via the `<f11>-R` key binding. No code in PEL changes cua-mode custom variables, and you are therefore able to activate the cua-mode the way you see fit. PEL promotes using the cua-rectangle-mark-mode by itself, not the remainder of the cua-mode. PEL provides several other keys you can use to perform cut and paste operations. This makes the cua-mode less attractive. See ∑ **Cut & Paste**.		
Toggle CUA mode On/Off	`M-x cua-mode`	(**cua-mode** &optional ARG)	Toggle Common User Access style editing (CUA mode). • With a prefix argument ARG, enable CUA mode if ARG is positive, and disable it otherwise • This normally activates CUA-compliant `C-c`, `C-v`, `C-u` and `C-z` keys. That can be modified via the value assigned to cua-enable-cua-keys.
Copy region to kill ring (in CUA mode)	`C-c`	(**cua-copy-region** ARG)	Copy the region to the kill ring. • With numeric prefix arg, copy to register 0-9 instead.
Cut region to kill ring	`C-u`	(**cua-cut-region** ARG)	Cut the region and copy to the kill ring. • With numeric prefix arg, copy to register 0-9 instead.
Paste last cut or copied region or rectangle.	`C-v`	(**cua-paste** ARG)	Paste last cut or copied region or rectangle. • An active region is deleted before executing the command. • With numeric prefix arg, paste from register 0-9 instead. • If global mark is active, copy from register or one character.
The following rows describe other binding while CUA-mode is activated			
Changing size of marked rectangle	When a cua-rectangle mark is active, the following cursor keys can be used to move the rectangle without affecting the text content. Unfortunately the horizontal cursor keys do not work in terminal mode. The PEL package provides the <f11>-[and <f11>-] keys to help.		
Move rectangle up	`M-<up>`	(**cua-move-rectangle-up**)	Move the entire rectangle overlay (not the content) up.
Move rectangle down	`M-<down>`	(**cua-move-rectangle-down**)	Move the entire rectangle overlay (not the content) down.
Move rectangle left	• `M-<left>` • `<f11>-[`	(**cua-move-rectangle-left**)	Move the entire rectangle overlay (not the content) left.
Move rectangle right	• `M-<right>` • `<f11>-]`	(**cua-move-rectangle-right**)	Move the entire rectangle overlay (not the content) right.
Select Rectangular region	• `C-RET`	(**cua-set-rectangle-mark**)	Start rectangle at mouse click position. • C-RET is not available in terminal mode (since there is no ASCII Control-Return)
	`<f11> . r`	(**pel-cua-rectangle-mark** &optional REOPEN)	Start CUA rectangle at mouse click position. • ⚠️🚧 This is experimental and currently does not work.
Cancel current rectangle		(**cua-clear-rectangle-mark**)	Cancel current rectangle.
Rotate in rectangle	`RET`	(**cua-rotate-rectangle**)	Moves the cursor to the next (clockwise) corner of the rectangle, so you can easily expand it in any direction over the cursor to the next (clockwise) corner of the rectangle, so you can easily expand it in any direction
Help for rectangle	`C-?`	(**cua-help-for-rectangle** &optional HELP)	

Activate CUA selection mode without the C-c/C-v/C-x/C-z bindings		(cua-selection-mode ARG)	Turn CUA rectangle selection mode on. Then:
			- `C-RET` turns it off
			- `<RET` to cycle corners:
			  - when at top: insert on all lines after the rectangle,
			  - when at bottom: insert on all lines before the rectangle
			- Other characters are inserted before/after the rectangle. `<backspace>` also works to delete, which can be used to un-indent!
			- (suspend-frame) is mapped to C-z but is also mapped to C-x C-z. So loosing C-z is not that bad.

CUA Mode - References

Topic & link	Notes
CUA Mode - CUA Bindings @ GNU Emacs Manual	The official description. Start reading here.
Cua Mode @ EmacsWiki	Overview of CUA-mode capabilities.
Shift a region or line in emacs	See second answer for description of the cua-selection-mode
Emacs Tip #26: CUA mode	Has a suggestion to use C-SPC instead of C-RET and a little function that can detect which one to use.
Decrease indentation on a couple of lines	Talks about C-x <tab> bound to indext-rigidly
Emacs Wiki - CUA Mode	
Youtube - cua mode - #1	Shows how to use CUA mode with a vertical rectangle area to write the same text on each line at the right then at the left of the rectangle. Shows how to use it in a HTML file.
Youtube - CUA Mode #2	Another video on the same topic, sound is a bit louder but the hindi accent is strong.