Drawing

Sea axis. E. Elizabello Contraine PLD. Design of elizabello (experimental property) Contraine PLD. Design of experimental property) Contraine PLD. Design of experimental property (experimental property) Contraine PLD. Design of experimental property (experimental property) Artist Mode Artist Mode Contraine PLD. Design of experimental property (experimental property) Contrained to experime	<u>Description</u>	<u>Keystroke</u>	Function	Note
Service Filtraciants Controller PF. Controller		However, I have not I		
**************************************	-	<f11> D <f1></f1></f11>		Open the local copy of the <u>Drawing</u> PDF file unless a command prefix (like C-u) was used. In that case it opens the Github-hosted file instead.
Commands	support.	<f11> D <f2></f2></f11>		
Section Sect		used in terminal mode for simple things.		
Selective mode **Ofer packed to piece best in residuation is not accompany, writing reflectivements and the contract mode. **Selective mode **Call by a positive mode is contracted by a positive mode in both and accompany to the packed by a positive mode in the contracted by a positive mode in the dividuation. The dividual is a positive mode in the contract in the dividuation. The dividual is a positive mode in the contract in the dividuation. The dividual is a positive mode in the contract in the dividuation. The dividual is a positive mode in the contract in the dividuation. The dividual is a positive mode in the contract in the dividuation. The dividual is a positive mode in the contract in the dividuation. The dividual is a positive mode in the contract in the dividual is a positive mode in the contract in the dividual is a positive mode in the contract in the	Toggle artist mode	<f11> D a</f11>	(artist-mode &optional ARG)	With argument ARG, turn Artist mode on if ARG is positive. Artist lets you draw lines, squares, rectangles and poly-lines, ellipses and circles
See side. Text Modes * cfile* b p Picture Mode Commands While in plutieur roads the following commands help to pose use in wave that help circhwain from at. The possible. For example to plus the victorial of commands and without to the mode previously without any possible of the possible of the victorial of commands. While in plutine mode the circle of the victorial of commands. While is plutine mode the circle of the victorial of commands. While is plut mode. Picture Motion The followord or commands and without the file of commands, which yes of £1 = as it is the case for all mode. Picture Motion The followord or commands while yes of £1 = as it is the case for all mode. Picture Motion The followord or commands and commands. While yes of £1 = as it is the case for all mode. Picture Modes and the file of commands, which yes of £1 = as it is the case for all mode. Picture Modes and the file of commands, which yes of £1 = as it is the case for all mode. Picture Modes and the file of commands while the declared or file of the declared or file of the case for all mode. Picture Modes and the file of commands and the declared or file of the declared or file of the case and the of the case of all mode. Picture Modes and the file of commands while the and the case of all mode. Picture Modes and the file of commands while the declared or file of the case and the of the case of all modes. Picture Modes and the file of case and the file of the case and the of the case of all modes. Picture Modes and the file of case and the file of the case and the of the case of all modes. Picture Modes and the file of case and the file of the case and the of the case of all modes. Picture Modes and the file of case and the file of the case of all modes and the case of all modes. Picture Modes and the file of the case of all the case of all modes and the case of all modes and the case of all modes and the case of all the ca	Picture Mode	shapes for tables or even <i>drawing</i> ASCII-art. This work well in both graphics and terminal mode.		
down or gings por noncontably loward the left without changing the input mode. This is very useful to laye preclargular shapes that can be useful made. More lated the analysis of samples of the lated of communic, sample yips of £1 = as in the case for all mode. Picture Motion The following 2 commands stated by layer £1 = as in the case for all mode. Nove left - C-c < pre> C-c < pre> C-c < pre> C-c < pre> (picture-movement-left)	•		(picture-mode)	
Move left C-c < c				
C = C < C C C C	Picture Motion			
C-c < right> C-c	Move left		(picture-movement-left)	₩ With PEL when pel-use-winner user option is t the C-c <left> is used by</left>
Nove down C-c sup C-c sup (picture-movement-down) Nove down after self-inserting character in Picture mode. Nove southwest (re) C-c ' (picture-movement-me & apotional ARG) Move up and left after self-inserting character in Picture mode. Nove southwest (re) C-c ' (picture-movement-me & apotional ARG) Move up and left after self-inserting character in Picture mode. Nove southwest (re) C-c ' (picture-movement-we & apotional ARG) Move down and right after self-inserting character in Picture mode. Nove southwest (rev) C-c ' (picture-movement-we & apotional ARG) Move down and left after self-inserting character in Picture mode. Nove westport Nove up and very up and very up and very up and very down and left after self-inserting character in Picture mode. Nove westport Nove up and very up and ver	Move right		(picture-movement-right)	₩ With PEL when pel-use-winner user option is t the C-c <right> is used by</right>
C-c Colombo C-c Colombo C-c Colombo C-c Colombo C-c Colombo Colo	Move up		(picture-movement-up)	Move up after self-inserting character in Picture mode.
Move northeast (ne) C−c² (picture-movement-ne &optional ARG) Move up and right after self-inserting character in Picture mode. Move southwest (sw) C−c² (picture-movement-se &optional ARG) Move down and left after self-inserting character in Picture mode. Move westborthwest (sw) C−c² (picture-movement-se &optional ARG) Move down and right after self-inserting character in Picture mode. Move westborthwest (www) C−u² (picture-movement-se &optional ARG) Move up and two-column left after self-inserting character in Picture mode. Move westbouthwest (wew) C−u² (picture-movement-se &optional ARG) Move down and two-column left after self-inserting character in Picture mode. Move in Picture Model C−u² (picture-movement-se &optional ARG) Move down and two-column left after self-inserting character in Picture mode. Move in Picture Model C−u² (picture-movement-se &optional ARG) Move down and two-column left after self-inserting character in Picture mode. Move in Picture Model Picture Model Picture Model Picture Model Picture Model Picture Model Move Move	Move down		(picture-movement-down)	Move down after self-inserting character in Picture mode.
Move southwest (av) C−c / (picture-movement-aw & optional ARG) Move down and left after self-inserting character in Picture mode. Move southeast (se) C−c / (picture-movement-aw & optional ARG) Move down and right after self-inserting character in Picture mode. Move washtorithwest (www) C−u C−c ' (picture-movement-mw & optional ARG) Move up and two-column left after self-inserting character in Picture mode. Move washtorithwest (www) C−u C−c ' (picture-movement-aw & optional ARG) Move down and two-column left after self-inserting character in Picture mode. Move in Picture Mode The following commands movement feely, even in "locif" space, past the end of the current line or past the last line in the buffer, extending the withsteapace as necessary and converting hard table to space when necessary. Move down * C−p (picture-move-up ARG) Move wentically up, making whitespace if necessary. Move down * C−p (picture-move-down ARG) Move wentically down, making whitespace if necessary. Move to column following last non-whitespace * C−E (picture-forward-column ARG) Move wentically down, making whitespace if necessary. Move if picture-forward-column ARG Move current picture move forward ARG - 1 lines first. * C−E * C−E * C−E (picture-m	Move northwest (nw)	C-c `	(picture-movement-nw &optional ARG)	Move up and left after self-inserting character in Picture mode.
Move southeast (se) C−c (picture-movement-se & optional ARG) Move down and right after self-inserting character in Picture mode. Move westnorthwest (wmw) C−u C−c ' (picture-movement-me & optional ARG) Move up and two-column left after self-inserting character in Picture mode. Move eastsoutheast (eee) C−u C−c ' (picture-movement-se & optional ARG) Move up and two-column left after self-inserting character in Picture mode. Move eastsoutheast (eee) C−u C−c ' (picture-movement-se & optional ARG) Move down and two-column left after self-inserting character in Picture mode. Move in Picture Mode The foliowing commands movement freely, even in "void" space, past the end of the current line or past the last line in the buffer, extending the withespace as necessary and converting that after an integration commands, but other available navigation commands in described in ™ Navigation. Move up C−p (picture-move-down ARG) Move vertically up, making whitespace if necessary. Move down • C-n (picture-move-down ARG) Move vertically down, making whitespace if necessary. Move to column following last non-whitespace c-c (picture-move-down ARG) Move vertically down, making whitespace if necessary. Move right • C-f (picture-move-down ARG) Move vertically up, making whitespace if necessary. Move right • C-f (picture-move-down ARG) Move vertic	Move northeast (ne)	C-c '	(picture-movement-ne &optional ARG)	Move up and right after self-inserting character in Picture mode.
Move westnorthwest (wmw) C-u C-c (picture-movement-nw &optional ARG) Move up and two-column left after self-inserting character in Picture mode. Move eastnortheast (ene) C-u C-c (picture-movement-aw &optional ARG) Move up and two-column left after self-inserting character in Picture mode. Move eastsoutheast (ese) C-u C-c (picture-movement-aw &optional ARG) Move down and two-column right after self-inserting character in Picture mode. Move in Picture Mode Reciliowing commands movement freely, even in "viciof" space, past the end of the current line or past the last line in the buffer, extending the whitespace as necessary and converting hard table to appaces when necessary. Whose down and two-column right after self-inserting character in Picture mode. Move in Picture Mode C-p (picture-movement-aw &optional ARG) Move worthead by two entractive properties as necessary. Move down and two-column right after self-inserting character in Picture mode. Move in Picture Mode The following commands overrides standard navigation more freely. Move the picture on the picture mode of the current picture in the buffer as the end of the current picture mode in the buffer as the end of buffer, stop there without enco. Move down - C-p (picture-move-down ARG) Move vertically up, making whitespace if necessary. With argument, move that many innes. Position poin	Move southwest (sw)	C-c /	(picture-movement-sw &optional ARG)	Move down and left after self-inserting character in Picture mode.
Move eastnortheast (ene) C-u C-c / (picture-movement-ne &optional ARG) Move up and two-column right after self-inserting character in Picture mode. Move westsouthwest (wsw) C-u C-c / (picture-movement-sw &optional ARG) Move down and two-column right after self-inserting character in Picture mode. Move in Picture Mode The following commands movement feely, even in "void" space, past the end of the current line or past the last line in the buffer, extending the whitespace as necessary and converting hard tabs to spaces when necessary. Move down and two-column right after self-inserting character in Picture mode. Move up C-p (picture-move-up ARG) Move down and two-column right after self-inserting character in Picture mode. Move down C-p (picture-move-up ARG) Move the current line or past the last line in the buffer, extending the whitespace in fecessary. Move down c-n (picture-move-up ARG) Move vertically up making whitespace if necessary. With argument, move that many lines. With argument, move that many lines. Move to column following last non-whitespace in picture mode. Position point after last pon-blank character on current line. With argument, move that many lines. Position point after last pon-blank character on current line. With argument, move that many lines. Propertional point lines from the line ton popular districts and point lines from the lines ton popular districts and power last lines fro	Move southeast (se)	C-c \	(picture-movement-se &optional ARG)	Move down and right after self-inserting character in Picture mode.
Move westsouthwest (wsw) C-u C-c / (picture-movement-sw &optional ARG) Move down and two-column left after self-inserting character in Picture mode. Move in Picture Mode The following commands movement freely, even in "void" space, past the end of the current line or past the last line in the buffer, extending the whitespace as necessary and converting hard tabs to spaces when necessary. The following commands movement freely, even in "void" space, past the end of the current line or past the last line in the buffer, extending the whitespace as necessary and converting hard tabs to spaces when necessary. Move up ' C-p (picture-move-up ARG) Move vertically up, making whitespace if necessary. Move down ' C-n (picture-move-down ARG) Move vertically down, making whitespace if necessary. Move to column following last non-whitespace with the properties of the column following a state non-whitespace with the properties of the column following the properties of t	Move westnorthwest (wnw)	C-u C-c `	(picture-movement-nw &optional ARG)	Move up and two-column left after self-inserting character in Picture mode.
Move eastsoutheast (ese) C-u C-c (picture-movement-se &optional ARG) Move down and two-column right after self-inserting character in Picture mode. Move in Picture Mode The following commands movement freely, even in "void" space, but other available navigation commands in described in ∑ Navigation. Move up • C-p • (picture-move-up ARG) Move vertically up, making whitespace if necessary. • With argument, move that many lines. Move down • C-n • (picture-move-down ARG) Move vertically up, making whitespace if necessary. • With argument, move that many lines. Move to column following last non-whitespace character • C-f (picture-end-of-line &optional ARG) Position point after last non-blank character on current line. • Inserting the without error. Move right • C-f (picture-forward-column ARG &optional INTERACTIVE) Move column after last non-blank character on current line. • Inserting the provided ARG - Inserting the without error. Move in direction of current picture motion. • C-c f (picture-backward-column ARG &optional INTERACTIVE) Move cursor left, making whitespace if necessary. • With argument, move that many columns. Move in direction of current picture motion. • C-c C-f (picture-backward-column ARG) Move cursor left, making whitespace if necessary. • With argument, move that many columns. Move in direction of current picture motion. • C-c C-f (picture-motion-reverse ARG)	Move eastnortheast (ene)	C-u C-c '	(picture-movement-ne &optional ARG)	Move up and two-column right after self-inserting character in Picture mode.
Move in Picture Mode The following commands movement freely, even in "void" space, past the end of the current line or past the last line in the buffer, extending the whitespace as necessary and converting hard tabs to spaces when necessary.	Move westsouthwest (wsw)	C-u C-c /	(picture-movement-sw &optional ARG)	Move down and two-column left after self-inserting character in Picture mode.
## Whitespace is necessary and converting hard tabs to spaces when necessary. These commands override standard navigation motion commands, but other available navigation commands in described in ▼ Navigation. Move up	Move eastsoutheast (ese)	C-u C-c \	(picture-movement-se &optional ARG)	Move down and two-column right after self-inserting character in Picture mode.
Nove down C-n C-n (picture-move-down ARG) Move vertically down, making whitespace if necessary. With argument, move that many lines.	Move in Picture Mode	whitespace as necessary and converting hard tabs to spaces when necessary.		
• <down> • With argument, move that many lines. </down>	Move up	_	(picture-move-up ARG)	
Switch ARG of the property of the provided ARG - 1 lines first. Insection of buffer, stop there without error.	Move down		(picture-move-down ARG)	
Septional INTERACTIVE With argument, move that many columns.	last non-whitespace	С-е	(picture-end-of-line &optional ARG)	With ARG not nil, move forward ARG - 1 lines first.
Move in direction of current picture motion C-c C-f (picture-motion ARG) Move point in direction of current picture motion in Picture mode. With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. Move in direction opposite to current picture motion. Move point in direction opposite of current picture motion in Picture mode. With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. Picture Mode Rectangle The following commands allow drawing rectangles in the buffer as well as copy & kill them as storing/restoring to/from registers. Praw rectangle around region C-c C-r (picture-draw-rectangle START END) Clear & save rectangle C-c C-k (picture-clear-rectangle START END) &optional KILLP) Clear and save rectangle delineated by point and mark. The rectangle is saved for yanking by C-c C-y and replaced with whitespace. The previously saved rectangle, if any, is lost. With prefix argument, the rectangle is actually killed, shifting remaining text. Clear rectangle C-c C-y (picture-clear-rectangle & soptional KILLP) Clear rectangle delineated by point and mark into REGISTER. The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Vank and overlay saved rectangle, specially killed, shifting remaining text. Yank and overlay saved rectangle aved by C-c C-k The rectangle is positioned with upper left corner at point, overwriting existing tex	Move right			
## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With arctangle arctangle to for unrent picture motion in Picture motion. ## With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. ## With arctangle arctangle delineated by point and mark. ## The rectangle is actually killed, shifting remaining text. ## The rectangle is positioned with whitespace. ## With prefix argument, the rectangle is actually killed, shifting remaining text. ## The rectangle is positioned with upper left corner at point, overwriting existing text. ## With prefix argument, the rectangle is positioned with upper left corner at point, overwriting existing text.	Move left			
to current picture motion **With ARG do it that many times. Useful for delineating rectangles in conjunction with diagonal picture motion. **Picture Mode Rectangle** The following commands allow drawing rectangles in the buffer as well as copy & kill them as storing/restoring to/from registers. **Draw rectangle around region** C-c C-r (picture-draw-rectangle START END) Clear & save rectangle** C-c C-k (picture-clear-rectangle START END) & optional KILLP) Clear and save rectangle delineated by point and mark. The rectangle is saved for yanking by C-c C-y and replaced with whitespace. The previously saved rectangle, if any, is lost. With prefix argument, the rectangle is actually killed, shifting remaining text. Clear reactangle** C-c C-w (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) The rectangle delineated by point and mark into REGISTER. The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Yank and overlay saved rectangle and overlay saved in REGISTER and replaced with whitespace. The rectangle saved by C-c C-k The rectangle saved by C-c C-k The rectangle is positioned with upper left corner at point, overwriting existing text.		C-c C-f	(picture-motion ARG)	With ARG do it that many times. Useful for delineating rectangles in conjunction
Draw rectangle around region C-c C-r (picture-draw-rectangle START END) Draw a rectangle around region. Clear & save rectangle C-c C-k (picture-clear-rectangle START END & Optional KILLP) Clear and save rectangle delineated by point and mark. • The rectangle is saved for yanking by C-c C-y and replaced with whitespace. The previously saved rectangle, if any, is lost. • With prefix argument, the rectangle is actually killed, shifting remaining text. Clear reactangle C-c C-w (picture-clear-rectangle-to-register START END REGISTER & Optional KILLP) Clear rectangle delineated by point and mark into REGISTER. • The rectangle is saved in REGISTER and replaced with whitespace. • With prefix argument, the rectangle is actually killed, shifting remaining text. Yank and overlay saved rectangle C-c C-y (picture-yank-rectangle & Optional INSERTP) Overlay rectangle saved by C-c C-k The rectangle is positioned with upper left corner at point, overwriting existing texts.		C-c C-b	(picture-motion-reverse ARG)	With ARG do it that many times. Useful for delineating rectangles in conjunction
Clear & save rectangle C-c C-k (picture-clear-rectangle START END & optional KILLP) Clear reactangle C-c C-w (picture-clear-rectangle START END & optional KILLP) Clear reactangle C-c C-w (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) Clear reactangle C-c C-w (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) Clear rectangle delineated by point and mark into REGISTER. The rectangle delineated by point and mark into REGISTER. The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Yank and overlay saved rectangle delineated by point and mark into REGISTER. The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Overlay rectangle saved by C-c C-k The rectangle is positioned with upper left corner at point, overwriting existing text	Picture Mode Rectangle	The following comma	ands allow drawing rectangles in the buffer	as well as copy & kill them as storing/restoring to/from registers.
Clear & save rectangle C-c C-k (picture-clear-rectangle START END & optional KILLP) Clear reactangle C-c C-w (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) Clear reactangle C-c C-w (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) Clear reactangle C-c C-w (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) Clear rectangle delineated by point and mark. Clear rectangle is actually killed, shifting remaining text. Clear rectangle delineated by point and mark into REGISTER. The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Yank and overlay saved rectangle delineated by point and mark. Clear rectangle delineated by point and mark into REGISTER. The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Overlay rectangle saved by C-c C-k The rectangle is positioned with upper left corner at point, overwriting existing text		C-c C-r	(picture-draw-rectangle START END)	Draw a rectangle around region.
**TART END REGISTER & optional KILLP)* The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text. Yank and overlay saved rectangle C-c C-y (picture-yank-rectangle & optional INSERTP) Overlay rectangle saved by C-c C-k The rectangle is positioned with upper left corner at point, overwriting existing texts.		C-c C-k		The rectangle is saved for yanking by C-c C-y and replaced with whitespace. The previously saved rectangle, if any, is lost.
rectangle • The rectangle is positioned with upper left corner at point, overwriting existing tex	Clear reactangle	C-c C-w		The rectangle is saved in REGISTER and replaced with whitespace.
Leaves mark at one corner of rectangle and point at the other (diagonally oppose corner.		С-с С-у		 The rectangle is positioned with upper left corner at point, overwriting existing text. With prefix argument, the rectangle is inserted instead, shifting existing text. Leaves mark at one corner of rectangle and point at the other (diagonally opposed)

Description	<u>Keystroke</u>	Function	<u>Note</u>		
Overlay rectangle saved in register	С-с С-х	(picture-yank-rectangle-from-register REGISTER &optional INSERTP)	Overlay rectangle saved in REGISTER. The rectangle is positioned with upper left corner at point, overwriting existing text. With prefix argument, the rectangle is inserted instead, shifting existing text. Leaves mark at one corner and point at the other (diagonally opposed) corner.		
Edit Tabular Data	The following commands help manage tabular data using tab.				
Move to next tab stop	<tab></tab>	(picture-tab &optional ARG)	Tab transparently (just move point) to next tab stop. • With prefix arg, overwrite the traversed text with spaces. • The tab stop list can be changed by C-c TAB and M-x edit-tab-stops. • See also documentation for variable 'picture-tab-chars'.		
Set tab stops according to context of current line	C-c <tab></tab>	(picture-set-tab-stops &optional ARG)	Set value of 'tab-stop-list' according to context of this line. This controls the behavior of TAB. A tab stop is set at every column occupied by an "interesting character" that is preceded by whitespace. Interesting characters are defined by the variable 'picture-tab-chars', see its documentation for an example of usage. With ARG, just (re)set 'tab-stop-list' to its default value. The tab stops computed are displayed in the minibuffer with ':' at each stop.		
Delete backward		(picture-backward-clear-column ARG)	Clear out ARG columns before point, moving back over them.		
Kill rest of line	• C-k • <f11> - e</f11>	(picture-clear-line ARG)	Clear out rest of line; if at end of line, advance to next line. Cleared-out line text goes into the kill ring, as do newlines that are advanced over. With argument, clear out (and save in kill ring) that many lines.		
Insert new line, leave point before it See also: <u>∑ Whitespace</u>	С-о	(open-line N)	Insert a newline and leave point before it. If there is a fill prefix and/or a 'left-margin', insert them on the new line if the line would have been blank. With arg N, insert N newlines.		

Drawing — References

Topic & Link	Notes
Poor Man's UML / Emacs Artist Mode and Ditaa Demo - Youtube video	Video demo of Emacs artist mode. Shows how to draw UML diagram.