(i) Introduction

A “mode” with respect to an interface is a state in which input (such as typing) does something other than what it would do if such state were not engaged (for example, “Caps Lock” is a mode, which, when engaged, causes typing to produce all upper case letters). In other words, an interface is “modal with respect to a given gesture when (1) the current state of the interface is not the user’s locus of attention and (2) the interface will execute one among several different possible responses to the gesture, depending on the system’s current state.”* The use of modes in software tends to introduce the possibility for user error and confusion. For this reason, the Cat’s text editing interface is non-modal with respect to all editing actions. The Cat’s text editing interface also exhibits a significant degree of “monotony,” a corollary of modelessness, which means that the user may only perform each action in one way, thus avoiding the distraction to the user of having to decide in which way to perform a specific action.

A “spring-loaded mode,” “pseudomode” or “quasimode,” like a mode, is a state of an interface in which input does something other than what it would do if such pseudomode were not maintained. Unlike a mode, however, which is “engaged” until it is explicitly disengaged, a pseudomode is “maintained” by the user’s holding down of a modifier key (such as ❮SHIFT❯ or ❮LEAP❯) and exists only for so long as the user holds down such modifier key. When the user releases the modifier key, the pseudomode ceases to exist and the interface returns to its usual function (such as the user’s typing entering characters into the text). The sensory feedback from holding down a modifier key to maintain a pseudomode prevents many user errors that would result from the use of a mode rather than a pseudomode.

The two ❮LEAP❯ modifier keys on the Cat’s keyboard provide an incremental search pseudomode, where, when the user holds down either of the two ❮LEAP❯ keys to maintain the “leap” pseudomode, any characters typed by the user cause the cursor to move—“leap”—to the next (or previous, depending on which ❮LEAP❯ key the user is holding down) match of the characters typed by the user. When the user releases the ❮LEAP❯ modifier key, the cursor’s position is set to the first character of the current match of the search pattern the user typed.

The leap pseudomode’s incremental search allows the user to construct the search pattern interactively with immediate and dynamic results. As the user adds characters to, or erases characters from, the search pattern, the cursor moves to the next-closest match of the current search pattern as the user modifies the search pattern. In other words, if the user erases a character from the search pattern, the cursor jumps back to the

* Raskin, Jef. *The Humane Interface.* Reading, Massachusetts: Addison-Wesley, 2000. 42.

previously shown match as if the user had never added that character to the search pattern, and *vice versa*.

When the Cat goes to sleep, tapping any character wakes it and, if the tapped key is a character key, that character is inserted into the text so that no keystrokes are lost. In this way, the Cat's sleeping is also non-modal. To wake the Cat without modifying the text, the user may wake the Cat by tapping a modifier key, such as a ❮SHIFT❯ or ❮LEAP❯ key. When the user wakes the Cat, or starts the Cat using a disk that had previously been used to record text, the Cat starts in the same state as it was before having gone to sleep or shut down, meaning the cursor position, highlighting and other items of state and appearance are reflected as they were before the Cat either went to sleep or shut down.

(ii) Terminology

"Text" is composed of "characters," which are the letters, punctuation marks, numbers, symbols, spaces and return characters (indicated on the screen, when highlighted, by a hooked arrow, "↵") inserted into the text by the user's typing. A "string" is two or more characters that occur consecutively within the text.

A "tap" of a key is the user's pressing and releasing of that key with no intermediating action, while "holding" a key means for the user to press that key and continuing to press it (hold it down) until being directed to release it (let it go).

"Forward" is the direction in which English text is read—to the right and down (toward the end of the text), while "backward" is the reverse—to the left and up (toward the beginning of the text).

To "leap" means to move the cursor by using the incremental search pseudomode, which is maintained by holding down one of the two ❮LEAP❯ keys, ❮←LEAP❯ (leap backward) and ❮LEAP→❯ (leap forward), to a target within the text matching the entered search pattern. A search "pattern" consists of the string the user types while maintaining the leap pseudomode. The "target" is the complete string to be matched by the pattern and the "target character" is the initial character at the beginning of such target. The cursor always lands on the target character after a leap. The user may highlight the string between the cursor location before a leap and the target character (*i.e.*, the cursor's position after the leap) by tapping both ❮LEAP❯ keys at the same time. Having just entered text, rather than used the leap function, the user may highlight the text entered since the last leap by tapping both ❮LEAP❯ keys at the same time.

There may be more than one "match" of a pattern within the text, but there is only one target, which is the specific match the user is looking for with a specific leap. While the user maintains the leap pseudomode, the pattern may be adjusted dynamically by the user's erasing characters from, and adding characters to, the pattern, or both.

A pattern may be re-used to leap to the next match of such pattern within the text (for example, if the first match of such pattern is not the target) by the user's tapping of one of the ❮LEAP AGAIN❯ keys, which are located on the front of the ❮LEAP❯ keys and are activated by the user's holding down a ❮USE FRONT❯ modifier key and tapping the appropriate ❮LEAP❯ key. As with leaping, there are both backward leap again (❮←LEAP AGAIN❯) and forward leap again (❮LEAP AGAIN→❯) keys, each located on the front of the respective ❮LEAP❯ key.

"Creeping" means to move the cursor one character at a time, either backward or forward, by tapping the ❮←LEAP❯ or ❮LEAP→❯ key, respectively.

The two "anchors" (there are always exactly two anchors within a text) are (1) the cursor location before a leap and (2) the cursor location after a leap. The two anchors determine the beginning and end of the string that would be highlighted if the user were to tap the two ❮LEAP❯ keys at the same time. After (1) a leap that has established a set of anchors, after (2) the next leap, the second anchor becomes the first anchor and a new second anchor is set, with the previous first anchor losing its significance. Using the ❮LEAP AGAIN❯ keys, however, only adjusts the location of the second anchor and does not set a new first anchor. This allows the ❮LEAP AGAIN❯ keys to expand or contract the string that would be highlighted by tapping both ❮LEAP❯ keys. Creeping forward moves the second anchor along with the cursor, but does not move the location of the first anchor, and creeping backward moves the first anchor along with the cursor, but does not move the location of the second anchor. This allows the user to adjust the second anchor forward or the first anchor backward by creeping forward or backward, respectively.

(iii) Cursor

Characters typed are always inserted at the insertion point, which is indicated on the screen by a blinking rectangle—the "cursor." The cursor blinks at a faster rate when the text in its current state has been saved to disk (in other words, the file is "clean" because the file shown on the screen matches the file saved on the disk). The cursor blinks at its normal (slower) rate when there have been changes made to the state of the text shown on the screen that have not yet been written to the disk (in other words, the file shown on the screen is "dirty" because it does not match the file saved on the disk). Moving the cursor by leaping or creeping, or changing the highlighted portion of the text, does not dirty the file.

A "cursor rebound" occurs when the cursor, during a leap, returns to its original location because the pattern entered by the user does not match any string contained within the text. The cursor may not move at all or may appear not to move at all, or may jump back to its original location after having moved. For example, when the user adds a character to the pattern that causes the pattern no longer to match any string contained within the text but there had previously been a match, the cursor will

rebound to its original location (there no longer being a match of such string within the text).

(iv) Highlight

The highlighted text is that portion of the text indicated on the screen by being inverted within the solid, non-blinking rectangle generally following immediately behind the cursor. The one character immediately preceding the cursor is the default highlighted text.

The *"Fundamental Cursor Rule,"* which always applies while editing text on the Cat, states that the cursor always indicates where characters will be inserted into the text by the user's typing while the highlight always indicates what portion of the text shown on the screen will be deleted by the user's tapping of the ❮ERASE❯ key.

When the cursor is wide, the user's tapping of the ❮ERASE❯ key erases the one highlighted character immediately preceding the cursor and moves the cursor backward to the position previously occupied by such character. If the cursor is narrow, as it is after a leap, the ❮ERASE❯ key erases the one character under the cursor and the text after the cursor shifts one character backward. This effectively performs a forward erase, meaning the ❮ERASE❯ key erases forward through the text as the user taps it, rather than backward. If the cursor is extended, all of the highlighted text is erased by the user's tapping of the ❮ERASE❯ key.

(v) Wide Cursor

The default cursor is the "wide cursor," meaning the highlighted text consists of exactly the one character immediately preceding the cursor. The cursor is wide if it has been moved to its current position within the text by the user's inserting of characters into the text, rather than by the user's leaping or creeping to such position.

(vi) Extended Cursor

If the highlighted portion of the text, having been extended by the user's leaping or creeping and then tapping both ❮LEAP❯ keys at the same time, consists of more than the one character immediately preceding the cursor, the cursor is called an "extended cursor." The portion of the text highlighted as part of the extended cursor will be deleted by the user's tapping of the ❮ERASE❯ key.

(vii) Narrow Cursor

A "narrow cursor" occurs when the highlighted portion of the text and the cursor converge on the same character after the cursor is moved by leaping or creeping. When the cursor is narrow, the user's tapping of the ❮ERASE❯ key erases forward through the text.

(viii) Separated Cursor

The cursor is called a “separated cursor” when the highlighted text does not immediately precede the cursor. The cursor separates from the highlight when the user has highlighted a portion of the text and then moves the cursor to another location within the text in advance of moving the highlighted text to that new location. The highlighted text rejoins the cursor and the cursor becomes an extended cursor with the moved text highlighted when the user completes the leap, completing the movement of the highlighted text to its new location within the text.

(ix) Inserting

Typed characters are inserted into the text at the cursor’s current location, with word wrapping, line endings and paragraphs reformatted automatically.

The user’s typing of characters when a portion of the text is highlighted does not erase or replace such highlighted text or any other portion of the existing text. Instead, when a portion of the text is highlighted and the user types a character, such character is inserted into the text after the highlighted text. The one way to erase a portion of the text is (1) to highlight the portion of the text to be erased and then (2) to tap the ❮ERASE❯ key. In other words, all typed characters are always inserted into the text when not in the user is not maintaining the leap pseudomode. This eliminates the potentially negative and unintended side effect of the user deleting off-screen highlighted text simply by typing.

For best results, before inserting characters into the text, the user should move the cursor by leaping or creeping to the character that should come immediately after the characters to be inserted.

The insertion of characters into the text cannot be reversed by tapping the ❮UNDO❯ key, but the user may erase any just-inserted text by (1) tapping both ❮LEAP❯ keys at the same time, which highlights the text inserted since the last time the user performed a leap, and then (2) tapping the ❮ERASE❯ key.

(x) Erasing

The ❮ERASE❯ key erases either backward or forward through the text depending on whether the user moved the cursor to its current location within the text by (1) typing or (2) leaping or creeping, respectively. The direction in which the ❮ERASE❯ key will erase is indicated by the cursor’s appearance. If the cursor is positioned on the highlighted character (in other words, the cursor is narrow), the ❮ERASE❯ key will erase forward. Otherwise, when the cursor is wide or extended, the ❮ERASE❯ key will erase backward. This is the *“Erase Rule.”*

The direction of the ❮ERASE❯ key's action is automatic. The cursor is always wide after typing and always narrow after leaping or creeping. Immediately after leaping or creeping, the cursor narrows and the ❮ERASE❯ key erases forward because, generally, if the user leaps to a location within the text to make a deletion, the user intends the deletion to begin with the beginning of the target and to go forward through the text. When the user types, the cursor is wide because the user generally intends to use the ❮ERASE❯ key in this context to erase backward through the text to correct mistakes or modify the recently typed text.

Text that the user has just erased may be brought back to its previous location by the user's tapping of the ❮UNDO❯ key, which leaves the previously deleted text, which has now been brought back, highlighted. The ❮UNDO❯ key, however, can only reverse the deletion immediately prior to the user's tapping of the ❮UNDO❯ key. Tapping the ❮UNDO❯ key twice returns the text to the state it was before the first tap of the ❮UNDO❯ key. That is, tapping the ❮UNDO❯ key twice in a row is the same as the user having not tapped it at all. This is the *"Undo Rule."*

If the cursor reaches either the beginning or the end of the text, the direction of the ❮ERASE❯ key's action reverses. In other words, if the cursor is narrow and it reaches the end of the text, it will change to a wide cursor so that the next (and subsequent) taps of the ❮ERASE❯ key will erase backward through the text. Similarly, if the cursor is wide and it reaches the beginning of the text, it will change to a narrow cursor so that the next (and subsequent) taps of the ❮ERASE❯ key will erase forward through the text.

(xi) Leaping

"Leaping" means the user's moving the cursor to another position within the text by performing an incremental search. Leaping is a pseudomode maintained by the user's holding down of either of the two ❮LEAP❯ keys: ❮←LEAP❯ (leap backward) or ❮LEAP→❯ (leap forward), both of which are located below the ❮SPACE BAR❯ and are meant to be operated by the user's thumbs. The two ❮LEAP❯ keys are used to move the cursor backward or forward within the text to the next match of the pattern, entered by the user, within the text in the user's chosen direction depending on which of the two ❮LEAP❯ keys the user presses to initiate the leap.

To move the cursor to a new location—the target location—either backward or forward within the text, the user holds down either the ❮←LEAP❯ key or the ❮LEAP→❯ key, respectively, and types the pattern meant to match the target. The cursor will move to the first instance of the pattern within the text in the direction chosen by the user on character-by-character basis as the user types the pattern while holding down the chosen ❮LEAP❯ key. The user may adjust the pattern dynamically, meaning the user may erase existing characters from the pattern and add

new characters to the pattern, in order to make corrections to the pattern or to make the pattern more specific.

Characters typed while the user maintains the leap pseudomode are used only for the leap pattern and are not inserted into the text. The user may exit the leap pseudomode, causing typed characters to be entered into the text again, by releasing the held-down ❮LEAP❯ key. While the leap pseudomode is maintained by the user, the user's tapping of either of the two ❮USE FRONT❯ keys activates the ❮LEAP AGAIN❯ key, allowing the user to move the cursor to successive matches of the pattern through the text in the direction of the user's original leap. The user's releasing of the ❮LEAP❯ key and then holding down either of the two ❮USE FRONT❯ keys while tapping either of the two ❮LEAP❯ keys allows the user to leap again in either direction, based on which ❮LEAP AGAIN❯ key the user taps.

When the cursor has reached the target and the user has released the ❮LEAP❯ key, the Cat will position the cursor on the first character of the matched pattern (the target character). The Cat positions the cursor on the target character, rather than the last character of the pattern, because the user knows when initiating the leap what will be the first character of the pattern, but may not know, or may not be able to control, what will be the last character of the pattern required for the pattern to match the target intended by the user.

If the text does not contain a match for the pattern, the cursor will either (1) not move, if no characters of the pattern match any portion of the text, or (2) rebound to its starting position within the text after the user adds to the pattern the character which causes the pattern not to match any portion of the text. To force the cursor to rebound to its starting position, the user may add characters to the pattern that could not possibly match any portion of the text (such as "qqq") or erase all of the characters from the pattern. By releasing the ❮LEAP❯ key and then tapping the ❮UNDO❯ key, the user may also cause the cursor to return to its starting position within the text before the user began the leap, although this is not a cursor rebound, but the undoing of the leap.

Leaping is circular, meaning that leaping forward will, when the search reaches the end of the text, circle back to the beginning of the text and begin searching from there forward to the cursor's starting point. Similarly, leaping backward is circular in the reverse direction. In other words, when leaping, if there are no (more) matches of the pattern in the direction of the leap's search, the leap will circle past the end (if leaping forward), or the beginning (if leaping backward), of the text and continue searching for a match of the pattern within the text until the search has reached the cursor's starting position within the text. This ensures that the user does not miss a match of the pattern within the text in the other direction from that of the leap's direction.

When leaping, lower case letters added to the pattern match both lower case and upper case letters (in other words, the leap's search is case

insensitive), while upper case letters (typed by the user's holding down of a ❮SHIFT❯ key in addition to the ❮LEAP❯ key) match only upper case letters (in other words, typing an upper case letter makes the leap's search case sensitive).

After moving the cursor by leaping or creeping, the user's tapping the ❮UNDO❯ key returns the cursor to its previous position within the text before the user began the leap. If the user is not currently holding down the ❮LEAP❯ key, the user's tapping of the ❮UNDO❯ key returns the cursor to where it was within the text before the user began the leap. If the user is still holding down the ❮LEAP❯ key, if the user wants to return the cursor to its previous position within the text before the user began the leap, the user must either let go of the ❮LEAP❯ key before tapping the ❮UNDO❯ key or must cancel the leap by (1) using the ❮ERASE❯ key to erase all of the characters of the pattern or (2) causing the cursor to rebound to its starting position within the text by adding characters to the pattern that make the pattern not match any string within the text. In either case, the cursor returns to its position within the text where the user initiated the now-cancelled leap.

If the next instance of the pattern (the instance found by the leap's search) is not displayed on the screen before the leap moves the cursor to such match, the Cat repositions the text on the screen so that the now-visible match is displayed in the vertical center of the screen.

(xii) Leaping Again

Leaping again means either of (1) the user's pressing of either of the ❮LEAP❯ keys and tapping of either of the ❮USE FRONT❯ keys or (2) the user's pressing of either of the ❮USE FRONT❯ keys and tapping of either of the ❮LEAP❯ keys. In either case, this causes the Cat to repeat the user's last leap.

If the user is maintaining the leap pseudomode by holding down either of the two ❮LEAP❯ keys, the user's tapping of either of the two ❮USE FRONT❯ keys without first releasing such ❮LEAP❯ key will cause the cursor to leap to the next instance of the current leap's pattern within the text. If the user has released the ❮LEAP❯ key used to initiate the last leap, the user's pressing of either of the two ❮USE FRONT❯ keys and then tapping either of the two ❮LEAP❯ keys activates the ❮LEAP AGAIN❯ key. In the second case, where the user has released the ❮LEAP❯ key used to initiate the last leap, the user may tap either of the two ❮LEAP❯ keys, allowing the user to perform a leap again in either direction within the text.

The Cat retains the last pattern used for leaping in memory making it available to the user for use again until the user leaps to a different pattern. The Cat's memory of the last pattern allows the user to leap to the next or previous instance of the pattern without having to re-enter it. To move the cursor again to the next instance of the pattern used by the current leap (meaning the user has not yet released the ❮LEAP❯ key), the

user need only tap a ❮USE FRONT❯ key. To move the cursor again to the next instance of the pattern used by the previous leap, there being no current leap, the user must press either of the ❮USE FRONT❯ keys and then tap either of the ❮LEAP❯ keys. Holding down either of the ❮USE FRONT❯ modifier keys while tapping another key activates the function shown on the front of such other key, in this case the ❮LEAP AGAIN❯ function.

In other words, once the user has moved the cursor by leaping, the user's tapping of either of the ❮USE FRONT❯ keys while continuing to hold down the ❮LEAP❯ key will move the cursor to the next instance of the pattern within the text. If the user has released the ❮LEAP❯ key after performing the last leap, the user's tapping of either of the ❮LEAP❯ keys while holding down either of the ❮USE FRONT❯ keys will move the cursor to the next instance within the text of the pattern used for the previous leap. Regardless of the direction of the initial leap, the user's tapping of either of the ❮LEAP❯ keys while holding down either of the ❮USE FRONT❯ keys allows the user to leap again in the user's chosen direction.

If the user holds down both either of the ❮LEAP❯ keys and either of the ❮USE FRONT❯ keys, having pressed them in either sequence, the user's typing of additional characters adds to, and the user's tapping of the ❮ERASE❯ key removes characters from, the search pattern being used for the leap again.

Leap again, like leap, is circular, in that if there are no more matches of the pattern within the text in the direction in which the user initiated the leap again, the leap again will circle past the end (if leaping again forward), or beginning (if leaping again backward), of the text and continue searching until the search has reached the cursor's starting position within the text. When the leap again reaches the cursor's starting position within the text, it will stop on that starting point before continuing its search. In other words, the leap again does not skip the starting point, but treats it as a "match," highlighting the starting point in turn like any other match within the text.

(xiii) Creeping

The user's tapping of either of the ❮LEAP❯ keys, without the user's holding down, or tapping, of any other key at the same time, advances the cursor one character forward in the case of the ❮LEAP→❯ key and one character backward in the case of the ❮←LEAP❯ key. The user's moving of the cursor in either direction one character at a time in this way is called creeping. Creeping should not be used as a substitute for leaping, leaping being more efficient in situations involving movements of more than just a few characters at a time. Creeping is, though, often helpful for small movements of the cursor within the text or for adjusting the highlighted portion of the text in either direction by a few characters.

If there is any currently highlighted text, the distances covered by creeping either forward or backward will be added to the highlight when the user taps both ❮LEAP❯ keys at the same time to bring back the highlight (the user's creeping in either direction having collapsed the highlight). In other words, creeping will unhighlight the currently highlighted text, but using the ❮LEAP❯ keys to bring back the highlight will now include any portion of the text added to the previously highlighted portion of the text by the creep movements. In this way, creeping adjusts the starting and ending anchors, but does not set new anchors.

When the cursor is extended, the user's creeping backward or forward moves the cursor within the text so that only the respective left or right end of the highlighted text is now highlighted. The cursor is narrow when collapsing to the left and wide when collapsing to the right. After collapsing the highlight in this way and continuing to creep, the user may adjust the beginning or the end of the highlighted text, with the anchor points being adjusted accordingly (assuming the first anchor point is not lost by the user's typing or leaping).

As with leaping, after moving the cursor by creeping, tapping the ❮UNDO❯ key returns the cursor to its previous location before the last creep.

(xiv) Highlighting

To highlight text, position the cursor at the beginning or the end of the portion of the text to be highlighted, setting the first anchor, and leap to the end or the beginning, respectively, of the portion of the text to be highlighted, setting the second anchor, and, before releasing the first ❮LEAP❯ key, tap the second ❮LEAP❯ key and then release the first ❮LEAP❯ key. In other words, before completing a leap by releasing the ❮LEAP❯ key, tap the other ❮LEAP❯ key and then release the initial ❮LEAP❯ key. Think of this as "looking forward" or "looking backward." If leaping forward using the ❮LEAP→❯ key, tapping ❮←LEAP❯ "looks back" and highlights the text back to the anchor where the forward leap began (and *vice versa* for leaping backward using the ❮←LEAP❯ key).

If the user does not tap the second ❮LEAP❯ key while still holding down the first ❮LEAP❯ key, after releasing the ❮LEAP❯ key the text covered by the last leap may still be highlighted by the user's tapping of both ❮LEAP❯ keys at the same time, but this portion of the text may be highlighted only until the user modifies the text or moves the cursor to another location within the text by leaping (in other words, only until the anchors are changed, at which point the portion of the text between the new anchors would be highlighted on the user's tapping of both ❮LEAP❯ keys at the same time).

The ❮LEAP AGAIN❯ keys allow text to be highlighted in the same way as the ❮LEAP❯ keys.

To unhighlight currently highlighted text, when the highlight is complete (neither ❮LEAP❯ key is currently pressed), tap the ❮←LEAP❯ key if the cursor is at the beginning of the highlighted text or the ❮LEAP→❯ key if the cursor is at the end of the highlighted text. Once the highlighted text has been unhighlighted, to rehighlight it, tap both ❮LEAP❯ keys at the same time.

To highlight the characters typed since the last leap or creep (or the last time the user tapped both ❮LEAP❯ keys at the same time to highlight a portion of the text), tap both ❮LEAP❯ keys at the same time. The user's tapping of both ❮LEAP❯ keys at the same time and then tapping the ❮ERASE❯ key erases the text the user has inserted since the user's last leap or creep (or since the user's last tapping of both ❮LEAP❯ keys to select a portion of the text).

After leaping within the highlighted portion of the text, when the user releases the ❮LEAP❯ key, the highlighted text collapses into a narrow cursor. The user's tapping of both ❮LEAP❯ keys at the same time to rehighlight the text will result in the highlighted text extending from the narrow cursor forward to the end of the previous highlight. Leaping through highlighted text does not, as long as the leap is not completed by releasing the ❮LEAP❯ key, unhighlight the text (the user's creeping, however, will cause the text to unhighlight because creeping is equivalent to completing a one character leap).

Pressing the ❮UNDO❯ key leaves the cursor, as well as the state of the text and any highlight, as they were before the user most recently highlighted a portion of the text.

(xv) Unhighlighting

At any time when the highlighted portion of the text covers more than the one character immediately preceding the cursor, the extended cursor may be collapsed into a wide cursor or a narrow cursor. Tapping the ❮LEAP→❯ key "unhighlights forward" by collapsing the highlighted text to the one character behind the cursor. This leaves a wide cursor but not an extended cursor and no text is erased by this operation. Similarly, tapping the ❮←LEAP❯ key "unhighlights backward" by collapsing the highlighted text in the opposite direction, leaving a narrow cursor on the character at the beginning of the previously highlighted text.

The ❮UNDO❯ key reverses the unhighlighting of the text, leaving it once again highlighted. In addition to the ❮UNDO❯ key, the user's tapping of both ❮LEAP❯ keys together, which highlights the last inserted portion of the text since the user's last leap and until a new leap or highlight begins, will rehighlight the most recently unhighlighted text.

(xvi) Moving

To move text, begin by highlighting the portion of the text to be moved. To move the highlighted text, leap the cursor to the location within the text to which the highlighted text is intended to be moved. The highlighted text follows the cursor as the user moves the cursor throughout the text and will be set in place when the user releases the ❮LEAP❯ key. The moved text does, however, stay highlighted after being moved so that the user may move it again without first having to rehighlight it. If the first match of the pattern is not the target, before releasing the ❮LEAP❯ key, the user may refine the pattern such that it matches the target or leap again (or start a new leap), but do not unhighlight the text until it has reached its correct new location. Unhighlight the moved text, which is still highlighted after the move, to complete the move operation.

When moving text, the highlighted text and the cursor separate. That is, when there is some text highlighted, the next leap moves only the blinking cursor, while the highlighted text remains in place. When the user releases the ❮LEAP❯ key responsible for moving the cursor, the highlighted text moves to the position of the cursor and the highlighted text and the cursor are once again adjacent as an extended cursor.

The *"Moving Text Rule"* states that for the most effective movement of text, the user should move the cursor using the ❮LEAP❯ key to the character that should come immediately after the highlighted text once the move has been completed. In other words, after highlighting the text to be moved, the user should leap to the character that should come immediately after the highlighted text once the highlighted text has been moved to its new location.

A leap within the highlighted portion of the text does not result in the movement of any text.

❮UNDO❯ reverses the movement of a portion of the text by returning the highlighted (and now moved) text to its former location within the text and the cursor to its previous position.

(xvii) Copying

Using the ❮COPY❯ key, as opposed to highlighting text and moving it by leaping the cursor to another location, duplicates the copied text after the duplicated text's position within the text. The ❮COPY❯ key highlights the text between the two anchors automatically, so an explicit highlight by the user's tapping of both of the ❮LEAP❯ keys at the same time is not necessary before the user taps the ❮COPY❯ key, but if there is any highlighted text when the user taps the ❮COPY❯ key, the user's tapping of the ❮COPY❯ key will duplicate that text. To copy text, the user should first set the beginning and end anchors of the string to be copied by leaping and/or creeping to the beginning and end of the string to be copied and

then the user should tap the ❮COPY❯ key. This highlights the relevant portion of the text and creates a duplicate of the highlighted text immediately after the highlighted text's location, leaving the duplicated text highlighted and ready to be moved by the user to its desired location within the text. There is no "cut" command where text disappears into an invisible clipboard. Likewise, the copy command places the duplicated text right after the text that was duplicated and not into an invisible clipboard.

Tapping the ❮UNDO❯ key erases the duplicate of the highlighted text, but only the most recent duplication.

(xviii) Scrolling

Because spaces and return characters are characters within the text just like any other, the user may use them in leap patterns to leap to spaces and return characters. To leap forward word-by-word, the user should leap to the next space and to leap forward sentence-by-sentence, the user should leap to the next full stop (or to the next occurrence of two adjacent spaces, if the user has separated each sentence with two spaces). The user may leap backward word-by-word or sentence-by-sentence in the same way. To leap forward (or backward) paragraph-by-paragraph (also known as "Cat-style scrolling"), the user should leap to the next (or previous) occurrence of two adjacent return characters by holding the relevant ❮LEAP❯ key while tapping the ❮RETURN❯ key twice and then releasing the ❮LEAP❯ key.

To scroll one line up or down at a time, the user may hold down a ❮SHIFT❯ key and tap the ❮←LEAP❯ key or the ❮LEAP→❯ key, respectively. The user may repeat scrolling one line at a time by using the ❮LEAP AGAIN❯ keys.

Leaping and then canceling the leap, whether the user cancels the leap by erasing all of the characters of the pattern, adding characters to the pattern to cause it not to match any portion of the text or by ending the leap and then tapping the ❮UNDO❯ key, allows the user to browse forward or backward to a point in the text and then returning to the starting point. It is also possible to place markers in the text by the user's inserting of characters that would not otherwise appear in the text, such as "qqq" and then leaping forward or backward using the marker as the pattern to return to the marker's location within the text.

(xix) Learning

Sequences of tasks, meaning sequences of keystrokes, that the user intends to perform repetitively may be recorded so that the Cat may replay the tasks without the user entering the sequence of commands each time. This is known as creating or recording a "macro." To teach the Cat a macro, press and hold a ❮USE FRONT❯ key, tap the ❮LEARN❯ key

and then tap a number (❮1❯ for example). Release the ❮USE FRONT❯ key and perform the sequence of keystrokes meant to be saved as the macro and, when finished, press and hold a ❮USE FRONT❯ key, tap the ❮LEARN❯ key and then release the ❮USE FRONT❯ key. The ❮1❯ key has now become a command that may be run by pressing a ❮USE FRONT❯ key, tapping the ❮1❯ key and then releasing the ❮USE FRONT❯ key.

Searching for and replacing a string that appears in more than one location in the text is an example of when the ❮LEARN❯ key may be useful. To replace one string with another throughout the text, press ❮USE FRONT❯ + ❮LEARN❯ + ❮1❯, leap to an occurrence of the string and replace it with the desired replacement text and then press ❮USE FRONT❯ + ❮1❯. The Cat will repeat the keystroke sequence until it reaches the end of the text or runs out of matches of the string to replace.

To store text for repeated pasting, highlight the text to be learned and press ❮USE FRONT❯ + ❮LEARN❯ + ❮digit❯ + ❮LEARN❯. To insert the phrase in the text, press ❮USE FRONT❯ + ❮digit❯.

The ❮UNDO❯ key only reverses the last individual operation performed by the ❮LEARN❯ key. In other words, the entire sequence of a macro's commands cannot be undone: only the very last action performed by the macro will be affected by the ❮UNDO❯ key.

(xx) About and License