

# Rich Text Editor with Logger Requirements 1.0

Author: Daniel Riegelhaupt

This document describes the first version of system requirements for a simple rich text editor.

For this specific assignment an extra requirement is the logging of key pressed and mouse movements.

## Terminology in this context:

- Text: the actual text as the user has typed it.
- Document: the text as it is rendered on the screen.
- Character /char : any screen printable character, this includes white space and line breaks
  - We will start with ASCII characters.
  - If possible support should be added for UNICODE.
- Line break: a character that specifically denotes that after it a new line should begin
  - NOTE that a new line on the document doesn't mean there is a line break in the text; this can also happen due to rendering.
- Style: the total of the attributes describing the text rendering: font type (a string with a valid font type name), font size (a positive integer value), font color (from standard names in rgb.txt <http://sedition.com/perl/rgb.html> or RGB values R/G/B each in 0-255 range), bold(true/false), italic (true/false) and underlined (true/false)
- Word: a sequence of characters that does not include a tab, white space or line break
- Alignment: the way words are ordered on a line, and sentences are cut/wrapped to fit within the document's margins.
  - The default (and for the moment only) alignment is: Flush left. (aka 'Left Alignment' or 'Ragged Right')
  - In future versions support might be added for Flush Right (aka 'Right Alignment' or 'Ragged Left'), Justified and Centered.
- Text cursor: the cursor indicating where we type.
  - The 'insert text cursor' : it can look like a (pulsating) |
  - The 'overtyping text cursor': it can look like a (pulsating) ■
- Mouse cursor: the mouse cursor.
  - The 'document mouse cursor', when the mouse is over an area in the document: it can look like I
  - The 'gui mouse cursor', when the mouse is over the gui: it is the standard OS version of the mouse so it can look like 
- Margin: the left and right vertical borders of the document.
  - In this version they are constant.
  - In future versions they might become movable.
- Buffer: place where copied or cut text is kept.
  - For the moment this is only for this program.
  - If possible the good case would be to use the OS buffer and still be able to keep style information within the program (and most of it outside)
- Line: a line in the document
- Beginning of the document: the top left corner of the document
- Beginning of a line : the left margin of a line.
  - NOTE that depending on the alignment used, from the user's point of view, the beginning might be the right margin. That 'beginning' should be referred to as right

margin; the beginning of the line will always refer to the left. An analogous remark is applicable to 'beginning of the document'.

- When the text cursor is at the beginning of a line it is at the left margin and to the left of the first character
- End of a line: the last character (the one most to the right) of a line.
  - When the text cursor is at the end of a line it is at the right of the last character of that line.
- End of the document: the end of the last line.
- Overtyping mode: a mode of operation where we replace the (empty) character indicated by the text cursor with the typed character.
- Inserting mode: a mode of operation where we insert the typed character in the text in a position left or right of the text cursor depending of the alignment used.
- Action: typing (including line breaks, backspace and delete), copy, paste, cut, a font style change.

## Use Cases:

### 0. The Program starts:

\*the current style is set at default values (like for example times new roman ; 12 ; black; not bold ; not italic ; not underlined)

\*the current alignment is set to flushed left.

\*there is an empty document available to be written on

\* we are in insert mode

\*the text cursor is the 'insert text cursor' and is set at the top left corner of the document

### 1. Character key (except for line break)is pressed:

\* if the mouse cursor is shown hide it

#### 1.1 There is **no selection**

##### 1.1.1 current **alignment** is **Flush Left**

###### 1.1.1.1 **overtyping mode**

\*overwrite the character(or empty space) on which the text cursor is with the new character in the current style.

###### 1.1.1.2 **inserting mode**

\* the character is written to the right of the text cursor in the current style

\* if a character is now after the right margin rearrange the document/line according to the flush left alignment algorithm.

\* the text cursor is moved to the right of the newly typed character

#### 1.2 There is a **selection**

\* remove the selection from the text

##### 1.2.1 current **alignment** is **Flush Left**

\* the text cursor is placed ,but not shown ,to the left of where the selection used to be

\* continue with 1.1.1.2

### 2. Backspace key is pressed

\* if the mouse cursor is shown hide it

#### 2.1 current **alignment** is **Flush Left**

##### 2.1.1 there is **no selection**

2.1.1.1 there is **no character to the left** of the **text cursor** in the text

\* Do nothing

2.1.1.2 there is **a character to the left** of the **text cursor** in the text

\* remove the character to the left of the text cursor

\* rearrange the line/document according to the alignment algorithm

#### 2.1.2 there is a **selection**

\*remove the selection from the text

\* rearrange the line/document according to the Alignment algorithm

\* the text cursor is placed ,but not shown ,to the left of where the selection used to be

\*the current style becomes the style of the character to the left of text cursor (if there are no more characters it stay as it was)

### **3. Delete Key is pressed**

\* if the mouse cursor is shown hide it

#### 3.1 current **alignment** is **Flush Left**

##### 3.1.1 there is **no selection**

3.1.1.1 there is **no character to the right** of the **text cursor**

\* Do nothing

3.1.1.2 there is a **character to the right** of the **text cursor**

\* remove the character to the right of the text cursor

\* rearrange the line/document according to the Alignment algorithm

##### 3.1.2 there is a **selection**

\*remove the entire selection

\* the text cursor stays where it was

\*the current style becomes the style of the character to the left of text cursor (if there are no more characters it stay as it was)

### **4. enter key is pressed**

\* if the mouse cursor is shown hide it

#### 4.1 there is a **selection**

\* delete the selected text.

##### 4.1.1 current **alignment** is **Flush Left**

\* the text cursor is placed ,but not shown ,to the left of where the selection used to be

\* continue with 4.2

#### 4.2 current **alignment** is **Flush Left**

\* add a line break to the right of the text cursor

\* put the cursor to the right of the newly added line break.

(On the document this means the text cursor is placed on the left margin of the newline)

### **5. Arrow key is pressed**

\* if the mouse cursor is shown hide it

\*if there is selected text ; deselect it; show text cursor

#### 5.1 **Up arrow**

##### 5.1.1 current line is the **first line of the document**

\*do nothing

##### 5.1.2 current line is **not the first line of the document**

###### 5.1.2.1 current **alignment** is **Flush Left**

5.1.2.1.1 there is **text** on the **line above** the current one

\*place text cursor on the line above; to the right of the character closest to the same vertical as the current text cursor position. If there are 2 choices (the distance is equal) choose the left one.

5.1.2.1.2 there is **no text** on the **line above** the current one

\*place the text cursor at the beginning of the line above

#### 5.2 **Down arrow**

**Analogous to UP** but wit below instead of above and last line instead of first line

#### 5.3 **Left arrow**

5.3.1 the **text cursor** is at the **beginning of the document**

\*do nothing

5.3.2 the **text cursor** is **not** at the **beginning of the document**

5.3.2 current **alignment** is **Flush Left**

5.3.2.1.1 we are at the **beginning of a line**

\*place the text cursor to the right of the last character of the line above.  
(this can also mean the beginning of the line if the line above is empty)

5.3.2.1.2 we are **not** at the **beginning of a line**

\* move the text cursor one place to the left

#### 5.4 **Right arrow**

**Analogous to LEFT** with end of line instead of beginning, below instead of above, and right instead of left.

Only **exception**:

5.4.2.1.1 we are at the **end of a line**

\*place the text cursor at the beginning of the next line (= the line below) whether the line is empty or not

\*the current style becomes:

- the style of the character to the left of text cursor, if there is one.
- the style of the first character, if we are at the beginning of the document
- it stays as it was, if there are no characters

### **6. Insert key is pressed**

6.1 we are in **insert mode**

\* we are now in overtype mode

\* text cursor becomes the 'overtyping text cursor'

6.2 we are in **overtyping mode**

\* we are now in insert mode

\* text cursor becomes the 'insert text cursor'

### **7. Copy command**

the command can be called by GUI or with the shortcut ctrl+c

7.1 there is **no selection**

\* this command CAN'T be used (this means the GUI is grayed out and the shortcut won't have any effect)

7.2 there is a **selection**

\*copy the selected text (style included) to the buffer

\* deselect the selected text

\*show the text cursor

### **8. Cut command**

the command can be called by GUI or with the shortcut ctrl+x

8.1 there is **no selection**

\* this command CAN'T be used (this means the GUI is grayed out and the shortcut won't have any effect)

8.2 there is a **selection**

\*copy the selected text (style included) to the buffer

\* remove the selected text

\* the text cursor is to placed, the right of the character that came before ( = to the left of) the selection that has just been deleted (this position is independent of the alignment used)

## **9. Paste command**

the command can be called by GUI or with the shortcut ctrl+v

### 9.1 the **buffer** is **empty**

- \* this command CAN'T be used (this means the GUI is grayed out and the shortcut won't have any effect)

### 9.2 the **buffer** is **not empty**

#### 9.2.1 current **alignment** is **Flush Left**

- \*insert the text in its own style at the right of text cursor
- \*place the text cursor to the right of the last character of the newly added text
- \*the current style becomes that of the last character of the pasted text
- \*rearrange the line/document according to the current alignment algorithm if necessary

## **10. Undo command**

the command can be called by GUI or with the shortcut ctrl+z

### 10.1 there is **no action** to be undone

- \* this command CAN'T be used (this means the GUI is grayed out and the shortcut won't have any effect)

### 10.2 there is **an action** to be undone

- \* Revert 1 action.

## **11. Repeat command**

the command can be called by GUI or with the shortcut ctrl+y

### 10.1 **no actions** have been done

- \* this command CAN'T be used (this means the GUI is grayed out and the shortcut won't have any effect)

### 10.2 there is **an action** to be redone

- \* redo that action

### 10.3 there are **no actions to redo but** there is an **action to undo**

- \* repeat the last performed action

## **12. one of font style attribute type, size or color is changed**

These attributes will be changed using a drop down menu in the GUI

### 12.1 **no selection**

- \*the current style attribute changes to the new value

### 12.2 **selection**

- \*the style attribute of the selection becomes that of the new value

- \*rearrange the text according to the current alignment algorithm if necessary ( not necessary when color was changed)

## **13. one of font style attribute bold , italic or underline**

These attributes will be changed using buttons in the GUI

### 13.1 **no selection**

- \*the current style attribute changes to the new value

### 13.2 **selection**

- \***no character** in selection has that **attribute on**

- \*turn that attribute on for the selection

- \***at least one character** has that **attribute on**

- \*turn that attribute off for the selection

\*rearrange the text according to the current alignment algorithm if necessary ( not necessary when the attribute was underline)

#### **14. left mouse button pressed and released (=clicked)**

\*if there was a selection, deselect it.

\*if it was in an area with characters calculate which character was clicked on or where the click happened

14.1 the **click** happened **above or on** the **last line** of the document

14.1.1 the **click** was **on a character**

\*the text cursor is placed to the left of that character

14.1.2 the **click** was on a part of a line with **no text**

14.1.2.1 current **alignment** is **Flush Left**

\*the text cursor is put to the right of the last character (or if the line is empty , the beginning of line)

14.1.3 the **click** was **between characters**

\*the text cursor is placed there

14.1.4 the **click** was **before** the **left margin**

\*the entire line is selected

14.2 the **click** happened **below** the **last line** of the document

\* do this use case as if it were happening on the last line

\*the current style becomes:

- the style of the character to the left of text cursor, if there is one.

- the style of the first character, if we are at the beginning of the document

- it stays as it was, if there are no characters

#### **15. left mouse button pressed and hold and dragged (and later on let go of)**

\*the text cursor disappears

\*the characters over which the mouse hovers are marked as selected (characters font color becomes white and they are marked in blue)

\*if an empty line is selected a vertical line is marked in blue.

\*if the mouse hovers back over a selected character it is unselected.

\*when the mouse is let go of

15.1 the **end point** of the **selection** is to the **left of the start point in the text**

(this means to the left or line above in the document)

\*the text cursor is placed to the left of the end point , but not shown

15.2 the **end point** of the **selection** is to the **right of the start point in the text**

(this means to the right or line below in the document)

\*the text cursor is placed to the right of the end point , but not shown

\*the current style is set to that of most left character in the text of the selection.

\*even though the style has deterministically been chosen if there are more than one values for style attribute color, size or type in the selection the drop down menu stays blank.

#### **16. save**

\* a save as dialog appears where user can save file

\* the file is saved on a server (that server can also be run locally)

#### **17. open**

\* an open dialog appears

\* the user can open

\*the document is opened

\*the text cursor is placed at the beginning of the document

\*the current style is set as the style of the first character

## **18. The program is closed**

18.1 there have been **no edits since last save**

\*close the program

18.2 there have been **edits since last save**

\* give the user the option to save before closing

\* once the user has decided (and if chosen: the save action preformed) close the program

## **Other Requirements:**

These are requirements that are more easily explained outside a use case context

### **Logging**

A separate concern/aspect is to log every key press and mouse click.

### **Cursor (mouse or text) movement/placement**

\*when moving/placing a cursor (either text or mouse): even if that cursor was previously hidden, it is shown again. unless it is explicitly stated that it should not be shown. ;

\*When the text cursor is not shown on the screen, the position that is remembered is its position in the text not that on the document

### **Style changes**

Whenever the current style changes the GUI dropdown menu of type, size and color will be updated.

### **The scrollbar**

\*Whenever after an edit the total height of the document increases add (or if it is already there Extend) the vertical scrollbar

\*Whenever after an edit the total height of the document decreases: shorten or completely remove the vertical scrollbar

\* when the scrollbar is moved: changed the view but the text cursor stays where it is

### **Undo/repeat**

As this is a bit difficult to explain using only words it will be explained using an example.

Consider this:

(P = present. this means the document consist of all actions taken from the beginning until and included the action before P , A# is an action)



\*if the user makes an action it is added:



\*the user can undo this action



\*the user can press undo again



\*if the user doesn't do an action but uses the redo command the a previously undone action is redone



\*the user can also redo A3, but if instead he does a new action (A4) every undone action that could be redone is removed.



\*in a situation where there are no undone commands to be redone if the user chooses the repeat command again the last performed action will be repeated.



\*when **saving any action to the right of the present** are **removed** from the list.

### Alignment Flush Left Algorithm

This tells us how text should be (re)arranged

When adding text (due to typing, pasting, line shifts because of line breaks or merging of words from two lines caused by delete and backspace ):

- \*if the line consist entirely off a word when the right margin of the line is reached:  
Simply cut off and continue adding/typing on the next line.
- \*if the line contains at least 2 words:  
Put the last word of the sentence on the next line and continue adding/typing from there

If due to adding in the next line words must be pushed apply this recursively until there is no need to continue on a next line.

Consider the following situation:

- Line a
- Line b
- ...
- Line c
- Line d
- ...

When due to an action like:

- \*removing characters in line a
- \* removing characters from the first word of line b
- \* splitting the first word of line b with a space
- \*removing the line break after line a

The first word of line b can now fit in line a do the following:

Move it there, and move as many sequential next words as possible then apply recursively on the lines below until one of the following conditions is met:

- \*the first word of line d can't fit on line c
- \*line c has a line break
- \*there are no (more) lines below