Here is my analysis of the problem of automatically renaming chord symbols when changing the fret of a fretboard diagram.

I assume that it is possible to recognize if to that note where our diagram is attached to there is also a chord symbol attached, too. Further I assume that the character of the key of the piece (of the score) can be recognized (whether it uses sharps or flats – or if it is completely absent which results in C-Major). At last I assume, that it is possible to recognize, how many frets a diagram has been moved by a keyboard or mouse action and the direction of that movement (a mouse action would always result in e movement by 1 fret, I guess).

At first glance it seems to come down to simply change an "E" into an "F" or an "Ab" into a "G" or something like that. It would indeed be as simple as that if there was not the possibility to input the intended fret directly into the fret-number box via keyboard. This makes it a bit more complicated but not so much that there wouldn't be a satisfying solution.

Because of the possibility of direct input via keyboard you have to calculate the number of frets to be moved and the direction (down or up). I guess this isn't really complicated if you are able to get the value of the fret box before and after direct input as well as before and after mouse action on one of the arrow-keys of the fret number box. Are you able to do so? "Value before" minus "value after" shows the number of frets moved and a positive value means moving upwards, whereas a negative value would mean moving downwards.

You could create a method for moving the chord symbols name just one "letter" up or down. That would be a "single-letter-move", i. e. a move by 1 fret.

Once you know in which direction to move and how many frets, you could iterate x-times the same process as for a single-fret-move. However, this leads to certain naming problems for the target symbol name.

Therefore iteration isn't a good idea, in my opinion. The following questions must be answered in advance:

I.   must the target symbol have an accidental or not, or has an accidental that existed in the original symbol's name to be omitted for the target name or not? (This depends from the number of frets to move. Simplified an even number of frets preserves the state of the original root name – if without an accidental, then the target name is also without – but when with an accidental, then the target name has the same accidental. An odd number of frets moved leads to adding an accidental when the original symbol's name had been without – or removing the existing accidental from the original symbol's name.)

II.   Do you cross the gap between "E/F" or "B/C" while finding the new root letter for the symbol? (Because then an accidental may have to be omitted that would otherwise be added according to the simple evaluation method.)

III. If an accidental has to be added, which one, a sharp or a flat?

IV.   Do you exceed the upper or lower boundary of note names of a scale while moving the fret? (This is dependent from the method for counting root letters for the chord symbol. Whether you use a limited list of root names or not, and if so, with which letter it starts and ends. For example, moving upwards by 5 frets from "A" lands you on "D". This proves to be problematic if your list is only from "C" to "B" but not if it is from "A" to

"G#". But moving downwards by 7 frets from "D" lands you on "G", whereas moving down by 3 frets from "A" would bring you to "F#").

All those sophisticated pondering can be circumvented, could even conveniently be solved by using an array of either 11 note names for one octave (and wrap the counting at the upper / lower end) - or an array of let's say 36 note names for all possible note names throughout the fretboard (without having to wrap the counting). So you just have to repeat the calculated number of steps on the fretboard within the array! Provided, of course, that you have initially retrieved the original name of the symbol and identified it in the array and set the step-pointer to it as a starting point. When the elements of the array contain already the complete note name, i. e. letter plus accidental, you just have to replace the original root letter in the symbol with that from the array where the pointer points to after having done all steps.

Now for the naming problems: The key in which a piece is set is usually reflected in the accidentals of the used chords. Harmonizing scale tones of that key results in chord names (we speak of the names only) that all bear the same accidentals - if at all. In the C-Major scale doesn't occur any chord with an accidental. From that can be derived, that pieces in C-Major usually contain no chords that bear an accidental in their names. In F-Major there is only one chord with accidental and that is a flat accidental. Other keys down the "cycle of fifth" add only more flats, i. e. when harmonizing scale tones of those key signatures, you'll find only chords with flat accidentals in their names but never with sharps. This holds true even when speaking of double dominants and the like, which are not necessarily scale-chords any longer. For G-Major it is vice versa. For complex pieces as classical ones there may sometimes occur chord names with flats in it along with chord names using sharps – if chord names explicitly written at all. Concerning diagrams – which are our topic – it is mainly about pop or jazz and for this purpose my claims made above would proof sufficiently valid.

I assume that it is easy enough to retrieve whether the key of the piece or the last key signature before the diagram we move by frets consists of flats or sharps. Hence we can define in advance whether the new name of the symbol should use rather flats or sharps if an accidental had to be added. If the key consists of sharps a changed symbol name should contain a sharp as well – if necessary, but never a flat. If the key consists of flats the changed symbol name should only contain flats, if necessary. With C-Major where the "key" consist of no accidentals, sharps shall be added if an accidental would proof necessary. This method makes you independent from the way the diagram together with its symbol has been stored in the template section. Enharmonic renaming takes place automatically according to the key of the piece when moving the diagram to the intended place on the fretboard.
Another method could be to decide the form of the accidental by the initial root name of the symbol. If it already has a flat, then flats are kept during the move, if it has a sharp already, sharps are kept. If it neither has a sharp nor a flat but is a "plain" letter, the form of the accidental is decided by the direction of the move. Downward moves result in flat accidentals, upward moves in sharp accidentals. But the former method looks more elegant in my opinion.

To be able to decide between sharps or flats at will, you need, in fact, two arrays, one with note names using flats only and one with note names using sharps only. Which array to apply depends on what is parsed from the key of the piece or – using the other method – from the accidental found in the original symbol's name or from the direction of the move respectively). For C- major no key

signature exists and it cannot be decided which array to apply. Then the array with sharps shall be applied.

<u>Now for the wrapping problem:</u> using a limited pair of array, say from "C, C#, D, D#, E, F, F#, G, G#, A, A#, B" and "C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B" respectively, makes it necessary to consider it as a circle and when counting upwards from "B" you should continue with "C" at the lower end. When counting downwards from "C" you should continue with "B" at the upper end. This method makes it simple to identify the "starting point" of the counting within the array because the original root name occurs only once in that array.

If you want to avoid wrapping, though, the arrays have to consist of about 3 octaves:
"E, F, F#, G, G#, A, A#, B, C, C#, D, D#, E, F, F#, G, G#, A, A#, B, C, C#, D, D#, E, F, F#, G, G#, A, A#, B, C, C#, D, D#" and
"E, F, Gb, G, Ab, A, Bb, B, C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B, C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B, C, Db, D, Eb". The middle octave represents the real fretboard and starts best with "E" to "D#"(lower E-string). Beneath that there is one more octave just for counting purposes. Moving a diagram from the 12th fret down then still leads to the appropriate root name even when identifying the starting point for counting at the first tone of the middle octave - and that without wrapping. The same is valid for the part of the array above the middle octave. Moving a diagram from the 1st fret up to the 12th while the root name is identified at the seventh half-tone step of the middle octave (B-string), as well leads to the correct root name. With this method you need not wrap, just count the steps for the pointer. However, you have to pay attention to where you identify the starting point for the counter, for every root name exists several times within the array - it has always to be the middle octave!

<u>Summary:</u> use two long arrays of 3 octaves, with complete tone names. Parse the character of the key signature - flat or sharp, which decides which array to use. Parse the actual Root name of the chord symbol combined with the diagram. Calculate the number of frets moved and the direction of the movement from the comparison of the fret number box before and after the keyboard/mouse action ("after" minus "before"). Identify the root name within the chosen array's middle octave (above the element 12) and place the pointer on it. Move the pointer in the array by the calculated number of steps into the recognized direction (negative means moving downwards). Retrieve the new root letter from the array's element where the pointer now points to and replace the old root name in the symbol with it. Done.

```
SYMBOL_NAME = get_chord_symbol_name();                          # cord symbol exists
if SYMBOL_NAME = "" then goto end_processing;                   # chord symbol doesn't exist
ARRAY_SHARP = "E, F, F#, G, G#, A, A#, B, C, C#, D, D#, E, F, F#, G, G#, A, A#, B, C, C#, D, D#, E, F, F#, G, G#, A, A#, B, C, C#, D, D#";
ARRAY_SHARP = "E, F, Gb, G, Ab, A, Bb, B, C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B, C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B, C, Db, D, Eb";
:start_processing;                                              # another mouse/keb. action?
KEY_SIGNATURE = get_key();                                      # retrieve last key signature
if pos("b", KEY_SIGNATURE) > 0 then FLAT = true                # consists of flats only
else SHARP = true;                                             # empty or consist of sharps
F = get_fret_number();                                          # initial fret of the diagram
If ((SYMBOL_NAME[2] = "b") or (SYMBOL_NAME[2] = "#")) THEN      # of root name contains accidtl.
   SOURCE_NAME = SYMBOL_NAME[1,2];                              # separate root name … (2 chars)
   REMAINDER = SYMBOL_NAME[3,12]                                # … from remainder (can't get > 12)
else
   SOURCE_NAME = SYMBOL_NAME[1];                                # ditto (but only 1 char)
   REMAINDER = SYMBOL_NAME[2,12]                                # dito
```

```
end;                                                        #
If (fretbox_mouse action() or keyboard_achtion()) then      # mouse or keyboard activity?
   M = get_fret_number();                                   # final fret of the diagram
   OFFSET = (M – F);                                        # number of frets and direction
   if focus_is_not_on_ diagram() then OFFSET =0;            # another object marked
   if OFFSET = 0 then goto end_processing;                  # done
   if FLAT then                                             # Use array with flats
      POINTER = pos(SOURCE_NAME, ARRAY_FLAT[12,12]);        # identify source root name
      POINTER = (POINTER + OFFSET)                          # identify target root name
      TARGET_NAME = ARRAY_FLAT[POINTER]                     # retrieve target root name
      TARGET_NAME = concat(TARGET_NAME, REMAINDER)          # complete target chord symbol
   else                                                     # use array with sharps
      POINTER = pos(SOURCE_NAME, ARRAY_SHARP[12,12])        # identify source root name
      POINTER = (POINTER + OFFSET)                          # identify target root name
      TARGET_NAME = ARRAY_SHARP[POINTER]                    # retrieve target root name
      TARGET_NAME = concat(TARGET_NAME, REMAINDER)          # complete target chord symbol
   end;                                                     #
   push_target_name_to_chord_symbol();                      # replace name of "chord symbol"
   SYMBOL_NAME = TARGET_NAME;                               # new "chord symbol" name in case
end;                                                        # another action takes place with
got start_processing                                        # fret of the same diagram
: end_processing                                            # all done
```

I'm not familiar with the programming style and the actually available functions of MuseScore. This Pseudo code is just to illustrate the steps that need to be taken. Syntax is of course not correct but should be understandable. The procedure is triggered only by a mouse or keyboard action on the fret number box and is dropped as soon as the diagram loses focus, i. e. as soon as another object is marked – or none.

I am convinced, after some brain juice has been spilled, the solution will look comparably simple, even elegant.

Limitations: I don't deny, that there may be situations when a user wants a certain diagram to be named in another way than the proposed procedure will result in. Then manual editing would be unavoidable. But these situations will be too rare, that they could challenge the advantage of the described method.

Best regards Michael, Gimmel