

Tony: a tool for melody transcription - Bug #890

Keyboard bindings on OSX not consistent (fn + vs cmd +)

2014-03-05 02:35 PM - Justin Salamon

Status:	Closed	Start date:	2014-03-05
Priority:	High	Due date:	
Assignee:	Chris Cannam	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>Shift octave = fn + arrows Cancel selection = fn + Esc Show/select candidates = cmd + return Switch candidates = cmd + arrows</p> <p>Would it be possible to have the first two use cmd + X instead of fn + X? This way the annotator doesn't have to constantly switch between cmd and fn.</p>			

History

#1 - 2014-03-05 06:07 PM - Chris Cannam

Note that because of the lack of working single-key shortcuts in Qt5 on OS/X at the moment, I'm going to continue to use Qt4 for official builds for the time being. (I'm making such a build now and will upload and ask for testing very shortly.)

The whole shortcut situation on the Mac is pretty baffling to me, but this is how I think it looks at the moment. I'm reading the Qt4 shortcuts straight from the screen here, as that's the build I have, and am filling in the rest from what you have told me.

I've included a suitably sarcastic "normal computer" column which lists the shortcuts on other platforms (Windows and Linux), and I've also included the other shortcuts (left over from SV) that might clash with desired Tony functions.

Function	Qt4 (official build)	Qt5 (dev build)	Normal computer
Show Pitch Candidates	Cmd+Return	Cmd+Return	Ctrl+Return
Pick Higher Pitch Candidate	Cmd+Up	Cmd+Up	Ctrl+Up
Pick Lower Pitch Candidate	Cmd+Down	Cmd+Down	Ctrl+Down
Clear Selection	Esc	Fn+Esc	Esc
Octave Shift Up	Fn+Up	Fn+Up	PageUp
Octave Shift Down	Fn+Down	Fn+Down	PageDown
Zoom In	Up	???	Up
Zoom Out	Down	???	Down
Speed Up	Cmd+Fn+Up	?Cmd+Fn+Up	Ctrl+PageUp
Slow Down	Cmd+Fn+Down	?Cmd+Fn+Down	Ctrl+PageDown

(Note that the last four are carried over from SV.)

So the question is what shortcuts do we want, and how do we get them cross-platform? I initially coded these:

Function	Shortcut
Show Pitch Candidates	Return
Pick Higher Pitch Candidate	Ctrl+Up
Pick Lower Pitch Candidate	Ctrl+Down
Clear Selection	Esc

Octave Shift Up	PageUp
Octave Shift Down	PageDown
Zoom In	Up
Zoom Out	Down
Speed Up	Ctrl+PageUp
Slow Down	Ctrl+PageDown

These are problematic on any platform because the higher/lower pitch candidate controls use Ctrl while octave shift etc do not. (This is similar to the complaint that opened this bug report.) It would probably be better to swap the pitch candidate controls with the zoom controls so that none of them used Ctrl.

However these turned out to be especially problematic on OS/X because

- There is no such thing as PageUp/PageDown keys (I was forgetting this -- is that what the Fn+Up/Down do generally?)
- The single-key shortcuts didn't work in builds with Qt5 (thus #867)

Looks like we have two options:

1. move away from single-key shortcuts and give all Tony editing controls a Ctrl (i.e. Cmd) modifier, or
2. move back to single-key shortcuts for Tony editing controls and insist that real work happens in Qt4-based builds until we have an upstream fix

It does seem a pity to be forced to change all the extant controls because of a bug in a particular library version on a single platform, especially since the single-key controls include such useful things as Play/Stop (on the space bar).

I would not be very happy to see Play/Stop, Clear Selection, the tool selection functions etc moved away from single-key controls on other platforms, but I do think we want cross-platform coherence where possible.

Either way, I guess we will also need an alternative to the PageUp/PageDown controls since Macs don't have those.

The first option might give us something like this (Mac Qt4)

Function	Shortcut
Show Pitch Candidates	Return
Pick Higher Pitch Candidate	Up
Pick Lower Pitch Candidate	Down
Clear Selection	Esc
Octave Shift Up	Fn+Up or Cmd+Up
Octave Shift Down	Fn+Down or Cmd+Down

(then we have to find new homes for zoom and play speed). An alternative for octave shift might be to make Up and Down perform octave shift operations if no alternate candidates are visible, or select among the candidates if they are -- though I'm not so keen on that offhand as I think it might be more error prone.

Going for the second option we might have

Function	Shortcut
Show Pitch Candidates	Cmd+Return
Pick Higher Pitch Candidate	Cmd+Up
Pick Lower Pitch Candidate	Cmd+Down
Clear Selection	Cmd+Esc
Octave Shift Up	???
Octave Shift Down	???

What are your thoughts?

#2 - 2014-03-05 08:33 PM - Justin Salamon

So, if I understand correctly, we need to make 2 decisions regarding tony key bindings:

1. single key or two-key combinations?
2. what combinations?

First, I think that regardless of the solution we choose for (1), we can keep the standard Tony commands single key. If everything is single key - no problem, if Tony commands use 2 keys, it's an easy way of distinguishing pitch editing functions from play/zoom controls. I.e., you know that whenever you have cmd pressed, you're changing the pitch curve. If command is not pressed, the pitch can't be changed. It might actually be a good safety measure to ensure people don't accidentally changed pitch values.

So, my view on (1) would be - both are fine, perhaps even 2-key bindings for pitch editing commands is a good thing?

Regarding (2), I actually quite like the idea of overloading the up/down keys (or cmd + up/down) - if you're in candidate mode they switch between candidates, if you're not in candidate mode they octave shift the pitch. This also prevents people from octave shifting whilst in candidate mode, which is very weird from an interaction point of view and should probably be avoided.

So, I guess my proposal would be:

Function	Shortcut			
Show Pitch Candidates	Cmd+Return			
Pick Higher Pitch Candidate	Cmd+Up [in candidate mode]			
Pick Lower Pitch Candidate	Cmd+Down [in candidate mode]			
Clear Selection	Cmd+Esc			
Octave Shift Up	Cmd+Up [outside candidate mode]		Octave Shift Down	Cmd+Down [outside candidate mode]

Rachel and I could alpha-test this by trying to actually create some annotations, so see whether this is comfortable or confusing.

#3 - 2014-03-05 08:35 PM - Justin Salamon

I may have not been completely clear in my previous update: by "tony commands" I mean commands we've added that specifically change the pitch curve. So, it would be single key for commands such as zoom, play, stop, etc., and two-key commands (cmd+X) for pitch editing functions.

#4 - 2014-03-06 01:55 AM - Rachel Bittner

I've been playing with annotating using the shortcut keys, and I agree with on both fronts -

for 1) I think having cmd + (single key) is kind of intuitive for editing the pitch curve.

for 2) i agree that overloading could be a nice solution, and also rather intuitive.

Side note, I see this issue is assigned to me - I can go make the changes, but should I? As you said it's a quick change, but I don't know if something special happens in creating the "official build" :)

#5 - 2014-03-06 09:23 AM - Chris Cannam

- Assignee changed from Rachel Bittner to Chris Cannam

- Priority changed from High to Immediate

The "official build" is just whatever source code is most recent in the repository, tagged, compiled, and packaged. The only magic bit is packaging the .app so that it will run on a different machine that doesn't have any of the developer libraries installed. Any changes you made would end up in the official build without any problem, as long as they'd been pushed.

So I would say yes, go ahead -- except that presumably you are all in bed now and I'm working on this again, so I might as well look into these now. If I get stuck on something I may assign it back again.

Here's what I'll do:

(ooh, I see Justin's table has some proper header markup that I didn't know about -- nice)

Function	Shortcut (in code)	Shortcut (expected for OS/X Qt5 build)	Note
Show Pitch Candidates	Ctrl+Return	Cmd+Return	a toggle control
Pick Higher Pitch Candidate	Ctrl+Up	Cmd+Up	octave-up if no candidates shown
Pick Lower Pitch Candidate	Ctrl+Down	Cmd+Down	octave-down if no candidates shown
Clear Selection	Ctrl+Esc	Cmd+Esc	abandoning any edits

And

Function	Shortcut (in code)	Shortcut (expected for OS/X Qt5 build)	Note
Zoom In	Up	Fn+Up	same as SV
Zoom Out	Down	Fn+Down	same as SV
Speed Up	Ctrl+PageUp	Cmd+Fn+Up	same as SV
Slow Down	Ctrl+PageDown	Cmd+Fn+Down	same as SV

It's not clear to me what to do with the (unmodified) Esc shortcut -- omit it, or leave it the same as SV, clearing the selection without abandoning the edits?

I'm also assuming that the remaining single-key shortcuts are kept single-key (e.g. play/stop) and we for now accept the fact that they aren't actually going to be single-key on the Mac.

About merging the octave-up/down and candidate-up/down controls, the problem I have with it is that it introduces "magic" at the expense of conceptual simplicity. I have many times written "magic" functions into my applications (I'm a bit of a sucker for it) and often it's proved to be a mistake.

Consider three levels of programmer-led UI design in terms of the "semantic gap" between program function and user conception:

1. Not even trying to bridge the semantic gap -- presenting UI functions that correspond directly to implementation mechanisms
2. Failing to bridge the semantic gap -- guessing the user's needs and providing functions that ("magically") reflect those guessed needs, but that do not actually map to a simple conceptual model for the user and so can't be understood in the case where the user fails to behave according to the guess
3. Bridging the gap -- producing UI functions that actually fit a simple conceptual model for the user

I have got a bit of experience at reaching the second of those, and it turns out that that isn't always helpful. Even the first can be preferable sometimes because at least the user can reason about what the computer intended. The third is much harder to achieve.

In this case my wariness is partly because I don't see how to label the corresponding menu functions, which is a bad sign. What does the proposed up/down control actually **do**?

However, rambling aside, I am not the intended user and my intuition is not very sound here, so I say fine -- let's test it and see. It might turn out to be just the thing.

#6 - 2014-03-06 02:49 PM - Chris Cannam

OK, I've made those changes (as of commit:e79f63161b41). For all my rambling about sinister dark magic, the merged shortcuts might well work quite well I think -- see what you think.

I've left the "clear selection" action attached to both Esc and Ctrl+Esc. One reason for this is that I discovered Ctrl+Esc is being grabbed by the window manager in the Linux desktop I'm using at the moment, so unless I bind it to Esc as well, I don't get to use it myself.

#7 - 2014-03-06 03:19 PM - Rachel Bittner

For "clear selection" - neither Esc nor Ctrl+Esc works on my machine (however fn+Esc does)

Everything else is working great, and I quite like the "magic" solution :)

#8 - 2014-03-06 06:42 PM - Chris Cannam

It may be that it's not possible to bind Ctrl/Cmd+Esc, or it may be that I've done it wrong.

I'm not at a Mac just now but if you search for Ctrl+Esc in MainWindow.cpp, you'll find that its action has two shortcuts attached, in a list, and the first in the list is the "primary" one.

The primary one (Esc in this case) is working (though in the guise of Fn+Esc on the Mac) but the secondary one apparently isn't. You might like to try simply swapping the order of the two just in case this is purely an artifact of the order in which they're listed. Let me know if you find it makes any difference.

#9 - 2014-05-21 02:41 PM - Matthias Mauch

- *Priority changed from Immediate to High*

Well, it's been three months now, so I change the priority from Immediate to High...

While this is working, I think the main issue is still that the key bindings on OSX are weird (as a result of Qt5 not supporting single-key combinations?).

Now I vaguely remember that Chris wanted to fix that in the actual Qt5 code, or we could try the workaround suggested here:

<https://bugreports.qt-project.org/browse/QTBUG-33015?focusedCommentId=238302&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-238302>

... which essentially means adding a toolbar button to make it work!

#10 - 2014-05-29 04:19 PM - Chris Cannam

Matthias -- just looking at the current build and I'm a bit baffled by the workaround you put in (to test the hypothesis about toolbar buttons). Can you explain what you changed and why?

I'm looking at the current commit:10e32085d109 in the Linux build.

The Edit menu contains the following. I've put the entries I don't understand in bold and added a note:

Text	Shortcut	Note
Undo	Ctrl+Z	
Redo	Ctrl+Shift+Z	
Navigate	1	
Edit	2	
Select All	Ctrl+A	
Esc	Esc	used to be labelled Clear Selection
Choose Higher Pitch	Ctrl+Up	
Choose Lower Pitch	Ctrl+Down	
Show Pitch Candidates	Ctrl+Return	
Del	Backspace	used to be Remove Pitches and Ctrl+Backspace
Split	/	used to be Ctrl+/ used to be Ctrl+\
Merge	\	
Form Note from Selection	Ctrl+=	
Snap Notes to Pitch Track		

Also there are four additional toolbar buttons labelled Esc, Del, Split, Merge -- I assume these are simply the visible face of the workaround in question.

Questions:

- Why have Clear Selection and Remove Pitches been renamed to Esc and Del in the menus? (Current hypothesis: The original names were too long to put in a toolbar button. Alternative hypothesis: Someone got mixed up between menu texts and shortcuts.)
- Why have the shortcuts for Remove Pitches, Split Note and Merge Notes had their Ctrl modifier removed? (Current hypothesis: To test the toolbar button workaround. Alternative hypothesis: They were not working before on the Mac for some reason, even with the Ctrl modifier.)

Adding a toolbar button is not a suitable long-term workaround -- though it is fascinating that it actually works! In the long term, it's pretty much essential that we actually get a fix for this as it's a blocker for the next release of SV as well as Tony. So I'm trying to work out how the land lies at the moment.

#11 - 2014-05-29 04:24 PM - Matthias Mauch

Chris Cannam wrote:

Matthias -- just looking at the current build and I'm a bit baffled by the workaround you put in (to test the hypothesis about toolbar buttons). Can you explain what you changed and why?

I'm looking at the current commit:10e32085d109 in the Linux build.

The Edit menu contains the following. I've put the entries I don't understand in bold and added a note:

Text	Shortcut	Note
<i>Undo</i>	<i>Ctrl+Z</i>	
<i>Redo</i>	<i>Ctrl+Shift+Z</i>	
<i>Navigate</i>	<i>1</i>	
<i>Edit</i>	<i>2</i>	
<i>Select All</i>	<i>Ctrl+A</i>	

<i>Esc</i>	<i>Esc</i>	<i>used to be labelled Clear Selection</i>
<i>Choose Higher Pitch</i>	<i>Ctrl+Up</i>	
<i>Choose Lower Pitch</i>	<i>Ctrl+Down</i>	
<i>Show Pitch Candidates</i>	<i>Ctrl+Return</i>	
<i>Del</i>	<i>Backspace</i>	<i>used to be Remove Pitches and Ctrl+Backspace</i>
<i>Split</i>	<i>/</i>	<i>used to be Ctrl+/ used to be Ctrl+\</i>
<i>Merge</i>	<i>\</i>	
<i>Form Note from Selection</i>	<i>Ctrl+=</i>	
<i>Snap Notes to Pitch Track</i>		

Also there are four additional toolbar buttons labelled *Esc*, *Del*, *Split*, *Merge* -- I assume these are simply the visible face of the workaround in question.

Yes.

Questions:

- Why have *Clear Selection* and *Remove Pitches* been renamed to *Esc* and *Del* in the menus? (Current hypothesis: The original names were too long to put in a toolbar button. Alternative hypothesis: Someone got mixed up between menu texts and shortcuts.)

Correct (current, and maybe alternative).

- Why have the shortcuts for *Remove Pitches*, *Split Note* and *Merge Notes* had their *Ctrl* modifier removed? (Current hypothesis: To test the toolbar button workaround. Alternative hypothesis: They were not working before on the Mac for some reason, even with the *Ctrl* modifier.)

Again, your first hypothesis is correct. Maybe also to simply highlight to an onviewer the fact that that's quite an essential thing to fix :)

Adding a toolbar button is not a suitable long-term workaround -- though it is fascinating that it actually works! In the long term, it's pretty much essential that we actually get a fix for this as it's a blocker for the next release of SV as well as Tony. So I'm trying to work out how the land lies at the moment.

Thank you!

#12 - 2014-05-30 05:47 PM - Chris Cannam

- File *testrun.log* added

Attaching test log file from OS/X 10.9 with Qt 5.3.0 with various extra debug output.

Here I started Tony, used *Cmd+R* to reload the last file, pressed 2 and 1 to (successfully) change the current tool, then pressed "f" to (unsuccessfully) try to zoom-to-fit, pressed *Backspace* (should have been inactive anyway) and then *Space* twice to (successfully) start and stop playback.

The remark in the earlier-linked bug report about the key-pair list in *qcocoahelpers.mm* not being sorted properly appears to be a red herring -- I added a few lines to report if the list is found to be out of order, and it seems OK (and the functions that use the list are returning the correct values).

#13 - 2014-05-30 06:25 PM - Chris Cannam

- File *testrun.log* added

Another log file with more debug output.

#14 - 2014-05-30 06:52 PM - Chris Cannam

Meanwhile, the workaround involving adding a toolbar button definitely is *not* a red herring:

```
QAction(0x10267a310) not connected to any widgets; won't trigger
```

from `correctActionContext` in `qshortcut.cpp`.

That explains why the 1 and 2 shortcuts work even though `f` does not (1 and 2 have corresponding toolbar buttons, and we know see *why* that makes them work, as well as having already seen *that* it makes them work). But why does e.g. `Cmd+R` work? It appears to go through a different flow, being handed off to the Mac menu implementation before ever reaching the `QShortcutMap` in this way.

[This page](#), though not about the same problem, does suggest another possible workaround (changing the shortcut context).

#15 - 2014-05-30 07:44 PM - Chris Cannam

Changing the shortcut context does not help.

And there is no dedicated logic rejecting these shortcuts -- the message saying it "won't trigger" is just reporting the state of things, because it's followed by a loop around all widgets associated with the action.

Meanwhile, for menu shortcuts that do work (like `Cmd+R`) the `handleKeyEvent` in `qnsview.mm` is never even called.

So we have a few angles:

- why does the single-key shortcut not get treated as a menu shortcut (and handled in the Cocoa menu layer before we reach `handleKeyEvent`)?
- why does the shortcut have no associated widget? (not even the `QMenu`)
- can we introduce a separate flow for shortcuts without widgets? (probably better not to hack Qt though unless we are providing a proper fix)
- as a last resort, presumably we can associate it with a widget but hide it

#16 - 2014-06-02 11:33 AM - Chris Cannam

Further digging: The single-key shortcuts **are** being set as Cocoa menu key equivalents in `QCocoaMenuItem::sync`. As far as I can see, that bit works just the same way as the shortcuts with modifiers -- it's just that Cocoa doesn't call back on them.

From references like e.g. <http://www.cocoabuilder.com/archive/cocoa/306819-menu-item-key-equivalent.html> it looks as if this might be an expected limitation in Cocoa.

Note that the single-key shortcuts that **do** work for us (e.g. 1 and 2) are working only because they have a toolbar button associated with them -- Cocoa doesn't invoke them through the menu either. (This contradicts the link above, on the face of it, but perhaps that developer also had toolbar buttons for zoom controls or something.)

#17 - 2014-06-02 12:23 PM - Chris Cannam

So to address those questions:

- why does the single-key shortcut not get treated as a menu shortcut (and handled in the Cocoa menu layer before we reach `handleKeyEvent`)?

We don't know that, but it doesn't appear to be a result of Qt failing to register it.

- why does the shortcut have no associated widget? (not even the `QMenu`)

Also a mystery. Querying `QAction::associatedWidgets()` after adding it to the menu shows 1 associated widget. The action in question is definitely the same one (same pointer value) both times, so the widget must have been detached at some point.

This may be something to investigate next -- it may help explain why this worked in Qt4 but doesn't work in Qt5, since presumably the Cocoa side (which naively appears to be at fault) hasn't changed between the two.

- can we introduce a separate flow for shortcuts without widgets? (probably better not to hack Qt though unless we are providing a proper fix)

Not a good idea I think, since that part of the code does not appear to be what's at fault.

- as a last resort, presumably we can associate it with a widget but hide it

#18 - 2014-06-02 03:33 PM - Chris Cannam

- *why does the shortcut have no associated widget? (not even the `QMenu`)*

Also a mystery. Querying `QAction::associatedWidgets()` after adding it to the menu shows 1 associated widget.

It **does** have an associated widget, and that is the `QMenu`. I was deceived by the debug output in `correctActionContext` in `qshortcut.cpp` -- I hadn't noticed that the same debug output appears in two different places in the function. I thought the first one was being fired, but actually it was the second (where the function has fallen through to checking for graphics widgets).

So the guilty commit appears to be this one:

<https://qt.gitorious.org/qt/qtbase/commit/bdebec4e2ef79f1771d2dcc22f9a919eb4487567>

It fixes a real bug, it seems:

<https://bugreports.qt-project.org/browse/QTBUG-30657>

I don't think there is much to be done about this in the Qt code. We can't add a check here that the menu is still enabled (following the logic in the comment added for that commit) because the "QPA level" enable/disable means that it acts on the Cocoa-native object rather than the Qt widget, and we don't have access to that here. And I don't know enough about Cocoa to know how to come up with any other fix.

I think we have to abandon trying to fix Qt and instead do one of the following:

1. Add a separate application-level shortcut that is not associated with the action (I mentioned earlier that changing the context on the action's shortcut didn't work, but a separate shortcut might)
2. Associate the shortcut with a hidden toolbar button as mentioned earlier

There is a fairly recently-accepted Qt bug about this issue at <https://bugreports.qt-project.org/browse/QTBUG-38256>. The report mentions a workaround using an application-level shortcut, which suggests that might be the way to go.

#19 - 2014-06-02 05:51 PM - Chris Cannam

OK, this should be fixed in commit:d1770a8ae921 using the application-level shortcut workaround.

#20 - 2014-06-02 06:23 PM - Matthias Mauch

Chris Cannam wrote:

| *OK, this should be fixed in commit:d1770a8ae921 using the application-level shortcut workaround.*

Awesome. Works here! Thank you!
(We/I could set to resolved, if you like.)

#21 - 2014-06-02 08:42 PM - Chris Cannam

- *Tracker changed from Feature to Bug*
- *Status changed from New to Closed*

Ah, it's OK, I can do it. I'll close it completely, we can always re-open this or another if there are any remaining problems. I wanted to change it to the bug tracker (rather than feature) as well, as that's pretty much how it turned out.

Downloads

testrun.log	15.4 KB	2014-05-30	Chris Cannam
testrun.log	18.2 KB	2014-05-30	Chris Cannam