

Sonic Visualiser - Feature #1158

true retina rendering on os/x

2015-03-11 05:07 PM - Chris Cannam

Status:	New	Start date:	2015-03-11
Priority:	Normal	Due date:	
Assignee:	Chris Cannam	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

Hi-dpi rendering already works fine on Linux and Windows, but OS/X is still problematic -- although textual elements are retina-ized, graphical rendering is still positively Minecraft.

This is because the view, and almost all layers, render onto pixmap backing store (at 1x resolution) before copying that to the widget (at what should be 2x, so ends up being unattractively scaled).

The fix is not as straightforward as I might hope. The view has a mixture of pixmap caching (for "scrollable" layers) and direct rendering (for layers that change depending on the position of the mouse).

We can generate a 2x pixmap for the scrollable cache and adjust the rect passed to the layer render() method accordingly, but that doesn't fix direct-renderers, and it doesn't help with the fact that layers will often call back on the view to establish its size and the mapping between (for example) x-coord and frame number. Also, this means that everything rendered to the scrollable cache ends up being scaled down, including text elements and the like, which we don't really want.

(Direct-renderers that actually render directly, using QPainter paths etc straight to the view, should be fine already. It's the direct-renderers that have their own pixmap caches, like the spectrogram, that are problematic.)

But conversely we can't just use a 1x cache because then any cached layers will end up low-res. Perhaps the very first thing we need to check is how far the view's scrollables cache is actually worthwhile in the first place -- which layer types benefit from it and by how much?

History

#1 - 2015-03-11 05:29 PM - Chris Cannam

Sorts of layer:

- Scrollable, cheap to render: TimeRulerLayer - could cache or render directly
- Scrollable, expensive, has own cache: WaveformLayer
- Scrollable, expensive, lacks own cache: TimeValueLayer in e.g. curve mode
- Non-scrollable, can be expensive, lacks own cache: TimeValueLayer in e.g. point mode
- Non-scrollable, expensive, has own cache: SpectrogramLayer
- Sometimes scrollable, expensive, has own cache: Colour3dPlotLayer

Many layers are sometimes scrollable, e.g. note layers are scrollable if not the top layer (so that local features are not being highlighted). I think much of the theoretical utility of the scrollable cache lies in managing the cost when a very large number of translucent layers is present on the same view, in which situation usually all but the top one will get cached. But this will fail as soon as they are placed over a non-scrollable background layer like a spectrogram, since the scrollable cache only caches as far as the first non-scrollable layer.

#2 - 2015-03-11 05:32 PM - Chris Cannam

... So we could probably get rid of the scrollable cache. The problem is that then each layer that has its own cache would have to do all its own scaling work, and that is complex.

#3 - 2015-03-11 05:43 PM - Chris Cannam

... So it may be better to ensure that layers always get a NxN-scale pixmap to render into, not the View directly. That means they have essentially the same behaviour as on other platforms, although they will need to scale fonts back up again (other platforms have proportionately bigger fonts everywhere when running in hi-dpi environments).

Then we could perhaps create a fake View-stub object with suitably scaled versions of dimension reporting and axis scaling functions to pass in to the layer render call?

Memory usage will be much higher though...

#4 - 2015-04-27 05:48 PM - Chris Cannam

- *File sv-retina-experimental.png added*

#5 - 2015-04-27 05:53 PM - Chris Cannam

- *File sv-nonretina.png added*

#6 - 2015-04-27 06:02 PM - Chris Cannam

Problems noted so far:

- Text rendered with drawVisibleText appears too small
- Much of the text in the vertical scale is too small
- The text in the time ruler is too small
- When mousing over a feature, the wrong feature is highlighted (although the right information is shown in the local feature identification box at top right)
- The zoom wheels are low-res
- It is noticeably slower to scroll

The layout is also very ugly in many other ways distinctive to the OS/X port, but this issue is about retina support only.

#7 - 2016-12-15 03:00 PM - Chris Cannam

The zoom wheels are still low-resolution, but everything else from that list is fixed.

#8 - 2016-12-15 03:03 PM - Chris Cannam

Another missing piece: when you export an image from the layer, you get non-retina resolution only.

#9 - 2016-12-15 03:32 PM - Chris Cannam

Zoom wheels are now retina (but the other thing is still outstanding).

Downloads

sv-retina-experimental.png
sv-nonretina.png

957 KB
852 KB

2015-04-27
2015-04-27

Chris Cannam
Chris Cannam