

Sonic Visualiser - Bug #1953

Wrong layer used for feature highlight when top layer is dormant

2019-10-16 09:36 AM - Chris Cannam

Status:	New	Start date:	2019-10-16
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

1. Open an audio file
2. Run a transform (in the same pane) such as Amplitude Follower
3. Switch off "Show" for the amplitude layer (but leave it as the top layer)
4. Wave the mouse pointer around in the pane

Expected behaviour: Features in the waveform are identified at top-right

Observed: Features in the invisible amplitude layer are identified at top-right

Is this bad? I think it might be. It's a little confusing, though I guess it could have its uses.

The Pane uses the "top layer" (View::getTopLayer) for several purposes:

- As the primary layer from which a vertical scale is drawn if possible (see kerfuffle in #1929 in Sonic Lineup)
- To determine whether to use a time x-axis (i.e. if not a slice layer)
- To configure the vertical panner
- To obtain the feature descriptions shown when waving the pointer around
- To measure and show measurement rectangles with the Measure tool
- To obtain and adjust extents when dragging vertically

Meanwhile it uses the "interaction layer" (View::getInteractionLayer) for these:

- To highlight (e.g. recolour) local points in-place in the layer when the mouse moves over them
- To snap selections
- For draw/erase/edit tool interactions
- For "zoom to region" interaction

The contention is that all of the purposes for which it is currently using the "top layer" should in fact also be using the "interaction layer".

The "top layer" is defined as:

| *the "top" layer in the view, whether visible or dormant*

The "interaction layer" is described as:

| *the topmost non-dormant (i.e. visible) layer in the view*

This means that the "interaction layer" is the layer that **appears** to be at the top, which is (probably?) what we want when displaying "top layer" information on the pane itself.

The one thing to be said in support of the current mechanism is that it's obvious from the property stack that there is a different layer active, even if it isn't visible. But the property stack itself is not always visible, so this is a slightly troubling mitigation.

We also have the "selected layer" (`View::getSelectedLayer`). This is quite different - it is effectively intended to return the "layer whose tab is topmost in the property stack", and it is the target layer for "external operations" like the Delete Layer menu function. If the current tab is that for a layer with the Show button switched off, then this will be an invisible layer; if the current tab is that for the pane itself, then it will be null.

History

#1 - 2019-10-16 09:37 AM - Chris Cannam

- *Description updated*

#2 - 2019-10-16 09:37 AM - Chris Cannam

- *Description updated*

#3 - 2019-10-16 09:39 AM - Chris Cannam

- *Description updated*