

Sonic Visualiser - Bug #1926

Display/value extents accessors in view classes are confusing and possibly contradictory

2019-09-25 09:31 AM - Chris Cannam

Status:	New	Start date:	2019-09-25
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

We have quite a mess of accessors and calculators for display/value extents (i.e. vertical extents) of layers:

Layer::getValueExtents (pv, implemented in individual layers)

- returns min, max, logarithmic, and unit
- typically returns actual extents of data in model, if meaningful

Layer::getDisplayExtents (pv, implemented in individual layers)

- returns min and max only (not unit or logarithmic)
- returns extents that have been set with setDisplayExtents, if any (for zoom and vertical scrolling)
- otherwise typically returns full value extents
- explicitly returns no-values in layers set to auto-align
- NB typically an awkward relationship with Layer::getCurrentVerticalZoomStep/setVerticalZoomStep

VerticalScaleLayer::getScaleUnits (pv, implemented in individual layers)

- also has getYForValue and getValueForY
- used by LinearNumericalScale and LogNumericalScale which actually draw scales

LayerGeometryProvider::getValueExtents (pv, implemented in View)

- returns min, max, and logarithmic
- takes unit as an argument, rather than returning it
- if any layers having that unit also have display extents (i.e. are not auto-align layers), returns the display (not value) extents of the lowest (??) such layer
- otherwise returns min and max of all the value extents offered by layers having that unit
- called when calculating y/value mapping in some layers when in auto-align mode

Pane::getTopLayerDisplayExtents

- returns all of value min/max, display min/max, and unit (but not logarithmic)
- returns extents taken directly from the top layer, not interaction layer or selected layer
- used from within Pane when dragging the pane up or down
- also has corresponding setTopLayerDisplayExtents
- was private, but I just made it public to call from MainWindow to support initialisation of layer scale and units when adding an empty box layer on top of a spectrogram (as it appears to be the only one of these that returns the unit)

Tangentially related:

VerticalBinLayer functions (pv, implemented in individual layers)

- getYForBin, getIYForBin, getBinForY, getIBinForY
- used by Colour3DPlotRenderer

DenseThreeDimensionalModel functions

- getBinValue, getBinValueUnit, shouldUseLogValueScale
- These were added when planning to unify the layer scales for FFT models (Hz assumed) and other colour 3d plots produced by plugins (e.g. ones that are also spectrograms)

History

#1 - 2019-09-25 09:31 AM - Chris Cannam

- *Description updated*

#2 - 2019-10-15 11:47 AM - Chris Cannam

I've slightly simplified the situation by, er, adding another method.

- I renamed LayerGeometryProvider::getValueExtents to getVisibleExtentsForUnit, and adjusted the implementation to return the display extents of the top-most (rather than bottom-most) visible (rather than any) layer that reports them for that unit.

The implementation still returns the union of all value extents for that unit if no layer returns any display extents. I don't think this is wise, it's definitely something to reconsider.

- I added a sibling View::getVisibleExtentsForAnyUnit for use when a layer (e.g. BoxLayer) can pick up any unit from the layers beneath it. This replaces the illicit external use of Pane::getTopLayerDisplayExtents (which was problematic in practice, see #1942).

This is in svgui commit:4f8c72adbf43, SV commit:95c082557ffd.

Note that Pane::drawVerticalScale duplicates some of the same logic as LayerGeometryProvider::getVisibleExtentsForUnit in order to decide which layer to ask to draw its scale. I think this is a problem - if the top layer is an auto-align one, then the scale appearing next to it should always (no?) be the scale of the layer that it is being auto-aligned to.