

QGIS Application - Bug report #13995  
singleband pseudocolor discrete rendered wrong

2015-12-15 02:23 PM - William Kyngesburye

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Category:</b>	Rasters	
<b>Affected QGIS version:</b>	2.8.3	<b>Regression?:</b> No
<b>Operating System:</b>	OS X	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b> end of life
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 22009

**Description**

I'm trying to display discrete colors for ranges of values, but it's using values below what's set for a range. The raster (attached) is continuous float values from 0 to 2, with a 2 decimal precision (.01). I want breaks at .01, .02, and .1. My color table is set as:

0	white	0 to .01 (essentially 0)
0.01	light grey	0.01 to .02 (essentially 0.01)
0.02	med grey	0.02 to 0.1 (or 0.02 to 0.09)
0.1	dark grey	0.1 to 2
2	dark grey	

See attached images for the correct and broken renderings.

Checking the raster values at the dark grey edge, I find that the .1 range really starts at .03, the next value up from .02 in the 2 decimal precision. The .02 range is OK presumably because for the precision it's the only value in that range.

I can workaround it by adding a .09 value with the same color as the .02 value.

I've tried scaling the raster by 100 so all values are integers, but this doesn't help.

**Related issues:**

Related to QGIS Application - Bug report # 14689: Singleband raster style ren...	Closed	2016-04-14
--	--------	------------

History

#1 - 2016-04-14 08:29 PM - Piers van der Torren

The problem here is that the behaviour of QGIS is different from what you expect. The behaviour is consistent, but the edges are defined different, like this:

```
0  white    <= 0
0.01 light grey  0 < .. <= .01 (essentially 0.01)
0.02 med grey  0.01 < .. <= .02 (essentially 0.02)
0.1  dark grey  0.02 < .. <= 0.1 (or 0.03 to 0.1)
2    dark grey  0.1 < .. <= 2 (or 0.11 to 2)
```

I've improved the automatic labels to make the ranges more clear, see <https://github.com/qgis/QGIS/pull/2979>

However, I noticed that the behaviour is not consistent with the gradient editor, which adds an extra class at the end, like you did.

If the type of breaks that you expected is also more common in other software, or otherwise more logical I think it makes sense to discuss if the renderer should be changed.

In the source documentation of QgsColorRampShader::exactColor() it says "Assigns the color of the lower class for every pixel between two class breaks.". That suggests the behaviour William expects actually also is the intended behaviour and the implementation is indeed wrong. Still the question is whether to fix the implementation or keep it and make it more clear what happens.

**#2 - 2017-05-01 01:06 AM - Giovanni Manghi**

- *Easy fix? set to No*
- *Regression? set to No*

**#3 - 2019-03-09 03:12 PM - Giovanni Manghi**

- *Resolution set to end of life*
- *Status changed from Open to Closed*

End of life notice: QGIS 2.18 LTR

Source:  
<http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/>

Files			
raster.tif	1.38 MB	2015-12-15	William Kyngesburye
color_correct.png	7.17 KB	2015-12-15	William Kyngesburye
color_broken.png	7.36 KB	2015-12-15	William Kyngesburye
color_broken.txt	269 Bytes	2015-12-15	William Kyngesburye
color_workaround.txt	313 Bytes	2015-12-15	William Kyngesburye