QGIS Application - Bug report #1963
"Ghost Lines" when using Anti-aliasing, polygons with shared boundaries, and no outline/QT::NoPen

2009-10-01 11:46 AM - springmeyer -

Status:	Closed		
Priority:	Low		
Assignee:			
Category:	Map Canvas		
Affected QGIS version:	master	**Regression?:**	No
Operating System:	All	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	wontfix
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	12023

Description

QT's anti-aliasing output exhibits an inherent limitation of AA, where polygon edges are drawn/feathered twice and the background color "bleeds" through what are supposed to be tightly shared edges of polygons.

I wonder if some of the new render hints in QT4.4 > might help work around this issue:

http://doc.trolltech.com/4.5/qpainter.html#RenderHint-enum

See also: http://trac.mapnik.org/ticket/428, where I am working on a workaround using AGG that may apply to the QT renderer, given an exposed API.

History

#1 - 2009-12-27 08:30 AM - Paolo Cavallini

Is this a QGIS, or a Qt bug?

#2 - 2009-12-30 08:19 AM - springmeyer -

QT limitation which may benefit from workarounds in QGIS. I think this is an important ticket to have open for other users to see, or potentially as an FAQ only if we don't find an easy solution.

#3 - 2010-09-19 07:11 AM - Paolo Cavallini

Is this still true for current Qt and QGIS versions? Please confirm.

#4 - 2010-09-21 12:57 PM - John Tull

This problem still exists in trunk with the most recent Qt release.

#5 - 2011-03-06 01:23 PM - Tim Sutton

Hi

I did a little test using both:

```
QPainter::SmoothPixmapTransform
QPainter::HighQualityAntialiasing
```

In src/gui/qgsmapcanvasmap.cpp for [[AntiAlias]] flags. Both choices did not resolve the issue for me. Using:

```
QGIS commit:e93c1841 (SVN r15735) Trunk
Qt 4.7
```

#6 - 2011-03-07 11:50 AM - springmeyer -

We solved this sufficiently in Mapnik within the AGG renderer by allowing the user to control the AA gamma. A gamma of 0 is aliased while 1 is fully anti-aliased and setting gamma to around .6-.7 is able to remove the faint lines while keeping an AA-ish look with much more definition to edges (like coastlines) than can be achieved by the only other known workaround: overpainting with a thin line of the same color as the polygon fill. In short, we solved this by reducing the aggressiveness of the AA algorithm so that polygons are slightly dilated rather are fully AA or aliased.

So, I assume that QT will expose somewhere an equivalent gamma setting (or partial AA ability) - as QT's renderer is originally based on AGG (http://labs.qt.nokia.com/2009/12/18/qt-graphics-and-performance-the-raster-engine/)

#7 - 2011-04-17 02:56 PM - Tim Sutton

- Status changed from Open to Closed

- Resolution set to wontfix

Reading the the referenced article it seems they only were inspired by AGG and didnt directly use any source code from AGG in their implementation. It seems like Qt4 rendering engine does not expose any agg-like gamma options and there isnt really any reasonable work around we can come up with for this. I am going to close this ticket since the only way to resolve this issue currently is to disable AA rendering it would seem and there isnt anything else we can do to fix it.

#8 - 2011-04-17 06:30 PM - springmeyer -

Okay.

#9 - 2013-02-26 01:24 AM - Henrik Uggla

- Assignee deleted (nobody -)

- Crashes QGIS or corrupts data set to No

- Affected QGIS version set to master

- Target version changed from Version 1.7.0 to Version 2.0.0

- Status changed from Closed to Reopened

- File testlinux.pdf added

Turning off "Make lines appear less jagged..." removes the thin lines on the screen but they still show when printing (see attached pdf). I can find no option for disabling anti-aliasing in the composer. Applies to both Windows7 and ubuntugis.

Note regarding attached pdf: The thin lines are displayed differently in different pdf-viewers.

#10 - 2014-06-28 07:37 AM - Jürgen Fischer

- Target version changed from Version 2.0.0 to Future Release - Lower Priority

#11 - 2017-03-27 09:22 AM - Johannes Kroeger

I realised that #7241 and #10166 were duplicates of this and closed them accordingly.

Has anything happened in the past years or maybe, hopefully, in QT5 that allows this to be fixed? It would highly improve cartographic quality in some use cases.

If not, a "use fill color for stroke" checkbox would be nice, if that indeed is a workaround for this (as suggested in the duplicates). It could be a useful addition for other use cases as well maybe?

#12 - 2017-05-01 12:58 AM - Giovanni Manghi

- Pull Request or Patch supplied set to No
- Regression? set to No
- Easy fix? set to No

#13 - 2018-09-01 01:38 PM - Johannes Kroeger

A workaround to this is:

Set the **Stroke width** to Hairline
Set the **Stroke color** to @symbol_color using **Data defined override**

I forgot who shared this, either Nathan or Nyall. :)

#14 - 2018-09-02 05:19 PM - Giovanni Manghi

- Status changed from Reopened to Closed
- Description updated

Johannes Kroeger wrote:

> *A workaround to this is:*
>
> *Set the **Stroke width** to Hairline*
> *Set the **Stroke color** to @symbol_color using **Data defined override***
>
> *I forgot who shared this, either Nathan or Nyall. :)*

is this in the docs? Can't find it.

#15 - 2018-09-30 03:55 PM - Johannes Kroeger

I don't think so, it was either on IRC or Twitter.

This issue remains, please keep it opened. The workaround is just that, a workaround!

Files

antialiasing_artifacts_with_nopen.png	54 KB	2009-10-01	springmeyer -
testlinux.pdf	330 KB	2013-02-26	Henrik Uggla