QGIS Application - Bug report #1262
Point and Multipoint symbols are not drawn centered at location

2008-08-30 09:35 AM - Steven Mizuno

Status:	Closed		
Priority:	Low		
Assignee:	Magnus Homann		
Category:			
Affected QGIS version:		**Regression?:**	No
Operating System:	All	**Easy fix?:**	No
Pull Request or Patch supplied:		**Resolution:**	fixed
Crashes QGIS or corrupts data:		**Copied to github as #:**	11322

Description

Point symbols are drawn approximately 2 pixels left and up from where they should be. I'm not quite sure of the amount, due to rounding errors in calculating the display location.

Multipoint symbols are drawn with the upper left corner of the symbol at the location of the point.

Expect that the symbol center would be at the point location.

This causes some frustration when using Identify, especially if the Search Radius is less than 0.5% or so.

attached is a 16x extract from a screen shot of examples - the blue square symbol is a POINT, the magenta square is a MULTIPOINT, three dark magenta lines start at the location the two point types are located at. I painted a black pixel at that location for reference. Also shown is an Identify highlight for the point/multipoint (both get the highlight as shown) centered at the location as expected.

Associated revisions

Revision 87816af1 - 2008-11-14 07:44 PM - Magnus Homann

Applying patch from smizuno to polish hard marker position accuracy. Many thanks! Fixes #1262

git-svn-id: http://svn.osgeo.org/qgis/trunk/qgis@9636 c8812cc2-4d05-0410-92ff-de0c093fc19c

Revision 19bc6b9b - 2008-11-14 07:44 PM - Magnus Homann

Applying patch from smizuno to polish hard marker position accuracy. Many thanks! Fixes #1262

git-svn-id: http://svn.osgeo.org/qgis/trunk@9636 c8812cc2-4d05-0410-92ff-de0c093fc19c

History

#1 - 2008-09-01 01:28 PM - Tim Sutton

Hi

Could you provide a small dataset I can use to replicate? I am interested to see if this patch will fix the point / multipoint alignment issues:

Index: ../src/core/qgsvectorlayer.cpp===

--- ../src/core/qgsvectorlayer.cpp (revision 9238)

++ ../src/core/qgsvectorlayer.cpp (working copy)

```
@ -3293,7 +3293,8 @
       transformPoint( x, y, theMapToPixelTransform, ct );
       //QPointF pt(x - (marker->width()/2),  y - (marker->height()/2));
       //QPointF pt(x/markerScaleFactor - (marker->width()/2),  y/markerScaleFactor - (marker->height()/2));
-      QPointF pt( x, y );
       QPointF pt( x*rasterScaleFactor - ( marker->width() / 2 ),  y*rasterScaleFactor - ( marker->height() / 2 ) );
+      //QPointF pt( x, y );
      #if defined(Q_WS_X11)
            // Work around a +/- 32768 limitation on coordinates in X11
```

#2 - 2008-09-03 05:38 PM - Steven Mizuno

data_for_1262.zip contains point.shp, multipoint.shp, linestring.shp, postgis_create_tables.txt. The [[PostGIS]] tables are what was used to show the point alignment problem screenshot. The shapefiles created from the [[PostGIS]] layers and also show the same.

The CRS for all is EPSG:26915. Use the filled square symbol of size 10.0 or so for the points. For the linestring use outline width of 0.00 which gives a 1 pixel line.

The point symbols should be centered at the convergence of the three lines.

#3 - 2008-09-05 05:39 AM - Steven Mizuno

On further testing I have found that at least some SVG symbols are drawn centered on the point location (within one pixel).

The internal symbols are the ones that are off by two pixels. I have also noticed that they appear clipped under some conditions. Observe the filled circle and square in the Point Symbol list. The left and top are clipped on my Windows installation.

#4 - 2008-10-18 09:24 AM - Magnus Homann
- Status changed from Open to In Progress

I did look into similar error two(?) years ago, and was then uable to draw the images correctly. Something wrong within Qt I believe. maybe it's fixed now, I'll have a look again.

Did you compiel it yourself, ans if so, which Qt version?

#5 - 2008-10-18 09:43 AM - Magnus Homann

It's coming back to me. The points are bitmaps which are referenced with the upper left corner. There is some fudging needed to find the delta-X and delta-Y from top left to center, and it is not reliable.

#6 - 2008-10-18 10:17 AM - Magnus Homann
- Status changed from In Progress to Closed
- Resolution set to wontfix

I think we can close this bug, multipoints are now drawn exactly like pointts, but not in dead center. Adjusting the outline width of the symbol changes the center placement of the symbol slightly. I don't think we can dio better as long as we use QImage?

#7 - 2008-11-13 06:10 PM - Steven Mizuno

- Status changed from Closed to Feedback

- Resolution deleted (wontfix)

I reopen this ticket because I have a patch for [[QgsMarkerCatalogue]] that draws the hard-coded point symbols more accurately than the existing one does.

I took this as a challenge as I believed that it was possible to draw the point symbols accurately, at least within one pixel.

The problems I found:

1. some calls (for circle and square) to QPainter::drawXXX() functions had two points passed in - these should have been upper left location and size; this is the main reason that the fudge factors had to be employed.

2. calls to QPainter::drawXXX() functions had float calculations, but the functions took integers - this causes whole pixel differences (left and up) due to truncation plus any other differences that occur elsewhere. This is why the symbols were clipped on the left and top.

3. the QImage may have an even number of pixels on a side, which makes it difficult to center.

What my patch does in imageMarker() and hardMarker():

1. makes the QImage have an odd number of pixels for width and height, so the center of the image can be placed properly. Also, the image width and height are odd for SVG rendering.

2. allows for the pen width, including Qt cosmetic pen (width=0, actually one pixel) when sizing QImage

3. uses floating point calculations, objects, and calls to the draw functions - QPointF, QPolygonF, QRectF, and such, leaving the conversion to integer pixel locations to the Qt draw functions.

4. the QImage size is passed to hardMarker() for centering the figure when drawing it. The hardMarker() function has an additional parameter. It is a private function, so the API is not affected.

5. provides proper rounding of float values to integer when centering the figure

6. pictureMarker() isn't currently used, I believe, so it is modified only so function calls are correct. Perhaps this function should be removed?

7. includes M_PI and DEG2RAD macros for some figure calculations

8. removes the iostream include as this interferes with the data stream operators used to send data into QPolygonF

Once I had it working I created three additional symbols (equilateral triangle, pentagon, regular star) using mathematical models, which were drawn centered as expected.

Now, symbols like the crosses are drawn on the map canvas with the point at which the lines cross at the location in question (within one pixel).

As submitted, filled figures are drawn at the size specified and then the outline is added - half inside, half outside the figure - as Qt's notion of outline. I provided some alternate code (the #if 0 block in hardMarker) to keep the overall symbol size at the specified size.

The patch is against commit:5fb69d1a (SVN r9612).

I am using Windows XP and Qt 4.4.0; also have tested on Linux / Qt 4.3.3

#8 - 2008-11-14 10:46 AM - Magnus Homann

- Status changed from Feedback to Closed

- Resolution set to fixed

Excellent! Many thanks!

Applied in commit:19bc6b9b (SVN r9637).

#9 - 2009-08-22 12:57 AM - Anonymous

Milestone Version 1.0.0 deleted

Files

Point_Multipoint_position_error.png	8.58 KB	2008-08-30	Steven Mizuno
data_for_1262.zip	2.98 KB	2008-09-03	Steven Mizuno
patch_for_1262.txt	11.9 KB	2008-11-13	Steven Mizuno