QGIS QT6 Meeting

30/11/2023

Attendees

- Alessandro Pasotti
- Nyall Dawson
- Julien Cabieces

Jitsi: https://meet.jit.si/QGIS-QT6

Topics

- Status of QGIS QT6 builds

Budget allocation

Task	Due date	Assignee	Budget
Find a solution and possible implementation for WebKit / WebEngine issue	2023-05-31	Nyall Dawson	5000
complete/fix issues related to PyQT + CI	2023-05-31	Julien Cabiécès	5000
Look into QT3D and QtWebEngine compatibility	2023-05-31	Martin Dobias	5000
Estimate packaging efforts			5000
Fix python tests when the bindings are available			0

Activities Reports

Julien:

- PyQt6 : PR went stale, no budget was consumed.
  How to get the PR merged? Split it?
  We can use the budget for reviewing.
  Use two separate folders for SIP 5 and 6 files.
  I will make the modification on the initial during december in order to make it easy to review and mergeable.
- Worked on PySide (task transferred from Denis with a separate budget), CI set and active, about 40 tests up and running. 80-90% of API wrapped. More details here : https://github.com/qgis/QGIS-Enhancement-Proposals/issues/237#issuecomment-1839008093

Nyall:

- There is still no movement from upstream Qt to officially support vectorised rendering of HTML from the webengine classes. Based on my experience with the development of Qt webengine and previous responses to this request, I deem it **extremely** unlikely that this functionality will ever be reintroduced.
- The most likely path forward is still to export HTML from the web engine classes to PDF, and utilise a 3rd party library to render PDF content to QPainter.
- I have tested this approach and can validate that HTML content exported from webengine to PDF retains full vector graphics (including text as text, and vectorised versions of tables/borders and other HTML elements). It is NOT rasterised content in a PDF! I've also tested and validated that we won't be held back by page size limitations caused by using the PDF format as a temporary container for rendered HTML (the Qt APIs allow us to export using any desired page size, so we will be able to continue to export at a fixed width and as a single page with height to match the content)
- The only library I can find for vector based rendering of PDF content to QPainter is https://github.com/JakubMelka/PDF4QT
- This is a GOOD library, and is well maintained. But it comes with shortcomings. Specifically:
  - It requires c++20 to build
  - It is Qt6 only
  - The library has many dependencies which are undesirable (and not required for our HTML exported PDF purposes, such as color management support and OpenGL based rendering)
- If we proceed with this approach, we have two options:

    - Heavily patch a copy of the library for QGIS' purposes. Allow it to build on qt5 and c++17, and rip out all the code we don't want. We'll end up with (another!!) heavily patched embedded library which has moved so far from upstream that it becomes effectively an unmaintainable/static snapshot of the library. This is a shame, given the rapid speed of development of the upstream library and the complexity of the PDF format (realistically, we will be dependent on upstream for bug fixes and can't just consider this a static mature library with no need to update)
    - Work with upstream to refactor their library to meet our needs, so that we can embed just a minimal version of their library without unnecessary build dependencies.
        - This is ongoing, and has been partially successful so far. A large amount of the work is complete, but some parts remain which I am trying to resolve with the upstream maintainer.
        - Noting that upstream will not fall back to Qt5 / c++ 17 support, so we will have to carry some downstream patches regardless to allow this. I have managed to get a working version of the library on c++17 and Qt5, and the patchset is **not** excessive.
- Based on recent interactions with the upstream library maintainer, they are receptive to our needs and open to the refactoring required for a minimal build. One red flag is that the maintainer seems reluctant to accept outside contributions and would prefer to do this work themselves, despite my offers of assistance.
- It's my belief that we will require one set of large changes to upstream to meet our needs (the minimal build refactoring), with very little to zero further upstream changes required following this initial work. Our needs are relatively modest, and we only need a fraction of the functionality from their library.
- In conclusion I think the best approach is to give the upstream library maintainer a couple of weeks to complete the refactoring, and reassess based on whether this work is completed.

- Julien: is there an option to implement the missing vectorization inside web engine? If possible we might ask KDAB.
    - Previous discussions with KDAB about this issue were unsuccessful – they indicated it was not something they were interested in working on (partly due to their lack of familiarity with the web engine codebase, and partly because they believed there was a very low chance of these changes being incorporated into the Qt classes)

Martin (via email):

Apologies, I was not able to fit the meeting into my schedule.

Generally, there is a PR (https://github.com/qgis/QGIS/pull/53255) which goes in the right direction, I still need to do a thorough review and testing of it to get it merged. Once merged, hopefully the Qt3D-QtWebEngine issues should be resolved.

On the financial side, I have used roughly a day on this (~1000 EUR) and I am hoping it should not need more than another day or two.

When: I would really like to get this done sometime in December...

Discussion about the roadmap

- PySide: target for 4.0
- Technology preview with PyQt6 before 4.0, to help users migrate plugins to Qt6