

Opis wymagań

Łukasz Szewczyk

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych

1 Wymagania

„Określenie wymagań, jakie musi spełniać oprogramowanie, jest tym miejscem projektu, w którym najbardziej i najwyraźniej stykają się interesy wszystkich jego udziałowców.” [1]

2 Wymagania funkcjonalne

2.1 Tworzenie wykresów

Biblioteka musi umożliwiać łatwe tworzenie wykresów 2D. Konieczne jest udostępnienie najpopularniejszych wykresów biurowych, czyli: słupkowego, liniowego i kołowego. Dodatkowo powinno być możliwe stopniowe uzupełnianie biblioteki o kolejne typy wykresów, takie jak: bąbelkowy, punktowy czy pierścieniowy. Powinno być możliwe tworzenie kilku wykresów w jednym układzie współrzędnych, np. słupkowy i liniowy. Biblioteka musi udostępniać takie elementy wykresów jak legenda, siatka, tło i osie, przy czym ich wyświetlaniem powinien sterować programista korzystający z biblioteki. Wszelkie właściwości wyświetlanych elementów reprezentujących próbki danych, takie jak kolor czy podpis, muszą być proste w obsłudze i przyjmować sensowne domyślne wartości.

2.2 Interakcja

Wykresy tworzone za pomocą biblioteki muszą być nastawione na interakcję z użytkownikiem, i tak dla wykresów osadzonych w kartezjańskim układzie współrzędnych powinno być możliwe zaznaczanie i przybliżanie ich fragmentów. Z kolei dla wykresów kołowego i pierścieniowego powinna istnieć opcja obracania ich. Wykresy muszą umożliwiać reakcję na ruchy kursora myszy po ich powierzchni, np. zawieszenie kursora na wycinku wykresu kołowego.

2.3 Interfejsy

Biblioteka musi umożliwiać tworzenie wykresów zarówno w klasycznym Qt jak i Qt Quick 2. API dla języka C++ powinno możliwie dobrze wpisywać się w styl Qt opisany w artykule [2].

2.4 Dane

Biblioteka musi zapewniać uniwersalny interfejs do danych oraz obsługiwać serie danych. Opcjonalna jest implementacja dla danych pobieranych w formacie XML lub JSON. Biblioteka musi umożliwiać łatwe dodawanie, modyfikowanie i usuwanie próbek danych.

2.5 Osie

Podstawową funkcjonalnością jest obsługa dwóch osi – jednej pionowej i jednej poziomej. Jednak implementacja nie powinna wykluczać obsługi większej liczby osi. Ponadto uniwersalność interfejsów powinna umożliwiać wprowadzenie w późniejszym czasie osi innych niż liniowa.

2.6 Efekty graficzne

Wszystkie opisane tu funkcjonalności są opcjonalne, a ich realizacja nie jest konieczna do zakończenia prac nad biblioteką.

Biblioteka może w jakiś sposób nadawać wykresom głębi i wrażenia przestrzenności. Kolejnym ciekawym dodatkiem, który podniósłby atrakcyjność wykresów jest wysokopoziomowy mechanizm motywów, podobny do *QStyle* [3]. Ostatnim elementem może być możliwość animacji procesu tworzenia wykresu – jest to element nadający aplikacjom dynamizmu. Łatwym powinno być generowanie plików graficznych na podstawie istniejących wykresów, np. w formatach .PNG i .SVG.

3 Wymagania pozafunkcjonalne

3.1 Przenośność

Biblioteka musi wpisywać się w politykę Qt brzmiącą: *pisz raz, kompiluj wielokrotnie*. Musi być przenośna na najpopularniejszych, wspieranych przez Qt platformach. Minimum to uruchomienie na platformach Windows oraz Linux (Ubuntu).

3.2 Struktura biblioteki

Biblioteka powinna mieć przejrzystą strukturę, która w połączeniu z dokumentacją w stylu Qt (przykład tutaj [4]) powinna umożliwić programistom sprawne przeanalizowanie jej działania i szybkie przystąpienie do tworzenia wykresów.

3.3 Wymienność biblioteki

Biblioteka powinna wykorzystywać mechanizmy pozwalające na tworzenie wymiennych bibliotek dynamicznych. Wprowadzenie nowej wersji biblioteki z niezmiennym interfejsem nie powinno wymagać przebudowania całej aplikacji.

3.4 Nowoczesność i uniwersalność

Biblioteka powinna wykorzystywać możliwie nowe technologie, m.in. Qt5. Jednak użycie standardu *C++11* nie jest wskazane ze względu na ograniczenie liczby potencjalnych odbiorców. Komponenty dostarczane do użytku programistom powinny być możliwie wysokopoziomowe i uniwersalne w użyciu.

Literatura

1. Inżynieria oprogramowania – rozdział 2, Krzysztof Sacha, Wydawnictwo Naukowe PWN, 2010, ISBN: 978-83-01-16179-8
2. API w stylu Qt <http://doc.qt.digia.com/qq/qq13-apis.html>
3. QStyle <http://doc.qt.digia.com/4.7-snapshot/qstyle.html#details>
4. Przykład dokumentacji Qt <http://qt-project.org/doc/qt-4.8/qobject.html>