

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б15-мм

Миграция проекта TRIK Runtime с Qt 5 на Qt 6

ГОЛОФАСТОВ Лев Дмитриевич

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
старший преподаватель каф. СП Я. А. Кириленко

Санкт-Петербург
2023

Оглавление

Введение	3
1. Обзор	4
1.1. Используемые технологии	4
1.2. Проблемы, которые необходимо решить	4
2. Постановка задачи	6
3. Реализация	7
3.1. Миграция проекта TRIK Runtime	7
3.2. Автоматическая сборка TRIK Runtime	9
Заключение	10
Список литературы	11

Введение

Хорошие программные продукты могут обеспечить длительное удовлетворение пользовательских потребностей в том или ином сервисе. Создание современного программного продукта не заканчивается на этапе разработки и внедрения. Эффективное сопровождение становится неотъемлемой частью продуктового жизненного цикла, включая исправление ошибок, предоставление поддержки пользователей и периодические обновления с внедрением новой функциональности.

Иногда компании прекращают поддержку предыдущих версий программного обеспечения при выпуске новых версий. Это решение продиктовано экономическими соображениями, так как поддержка устаревших продуктов может быть слишком дорогостоящей для их владельцев.

Например, рассмотрим Qt — это фреймворк для создания кроссплатформенных приложений на C++. В Qt 5.5 модуль Qt Script был объявлен устаревшим, и в Qt 6 он был полностью удален. Однако проект TRIK Runtime, который является частью проекта TRIK Studio, все еще использует этот модуль. В результате, после обновления до Qt 6, проект TRIK Runtime не будет работать.

TRIK Studio — среда программирования, позволяющая решать задачи как с помощью последовательности картинок, так и сложного текстового языка. TRIK Studio используется для преподавания в кружках робототехники, на уроках информатики, для проведения соревнований по робототехнике как в России, так и за рубежом.

На данный момент сборка проекта TRIK Studio осуществима на версии Qt 5.12. Однако рано или поздно Qt 5 перестанет поддерживаться. Главная задача заключается в миграции проекта TRIK Studio с Qt 5 на Qt 6, что позволит легче сопровождать его в будущем. В рамках данной учебной практики предлагается мигрировать проект TRIK Runtime с Qt 5 на Qt 6.

1. Обзор

1.1. Используемые технологии

Поскольку запланировано выполнить рефакторинг уже существующего кода, то та часть технологий, которая будет использована в данной работе, уже была задействована при создании проекта TRIK Studio [1].

- C++ — компилируемый, статически типизированный язык программирования общего назначения. На нем написана большая часть логики проекта TRIK Studio;
- Qt [2] — фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++. В рамках данной работы будет использована актуальная версия Qt 6, поскольку именно на эту версию запланировано мигрировать проект TRIK Studio.
- GitHub Actions [3] — это облачная платформа CI/CD, позволяющая автоматизировать конвейер сборки, тестирования и развертывания. Есть возможность запускать необходимые workflows на различных операционных системах (Linux, Windows, macOS), используя виртуальные машины, предоставляемые GitHub. Кроме того, GitHub Actions можно интегрировать с другими функциями GitHub, такими как Issues, Pull Requests, Push, GitHub Pages и др. workflows определяются с помощью YAML-файлов, которые содержат набор задач. В публичных репозиториях сервис можно использовать бесплатно.

1.2. Проблемы, которые необходимо решить

С выходом Qt 6 произошли изменения в API. Некоторые классы были добавлены, другие удалены, а также проведен рефакторинг, включая упрощение и переименование. Некоторые устаревшие компоненты,

такие как Qt Script, были удалены. Некоторые классы имеют аналогичную реализацию в Qt 6 под другим именем, но не всегда они совпадают напрямую. Например, функциональность класса QMediaContent из Qt 5 была разделена и интегрирована в различные классы модуля Qt Multimedia.

TRIK Studio представляет собой крупное приложение и набор небольших проектов, которые разработаны с использованием Qt. На данный момент они собираются с использованием версии 5.12. Учитывая, что основной проект зависит от своих подпроектов, имеет смысл начать с миграции этих приложений на Qt 6 поочередно, а затем удостовериться, что они успешно компилируются под Qt 6.

Взглянув на одну из главных компонент TRIK Studio проект TRIK Runtime [4], стало понятно следующее: данный проект зависит не только от некоторых модулей самого Qt, от зависимости от которых нужно избавиться (в частности, Qt Script), так еще и от библиотеки TRIK PythonQt [5], которая обеспечивает обёртку для языка программирования Python в Qt. Реализации обёртки для Qt 6 в этой библиотеке нет, что само по себе является довольно трудной задачей. Было принято решение отключить Python, убрав зависимости с PythonQt, и мигрировать TRIK Runtime на Qt 6 сначала без него. Этим и ограничена учебная практика в текущем семестре.

На данный момент есть следующие проблемы:

- TRIK Runtime не компилируется на Qt 6.
- Нет автоматизированной сборки для проекта TRIK Runtime на Qt 6.

Итак, на основании вышесказанного необходимо формализовать постановку задач.

2. Постановка задачи

Целью работы является миграция проекта TRIK Runtime с Qt 5 на Qt 6. Для её выполнения были поставлены следующие задачи:

1. Адаптировать исходный код проекта TRIK Runtime под Qt 6, обеспечив сохранение существующей логики и успешное прохождение имеющихся юнит-тестов.
2. Добавить автоматическую сборку на Qt 6 для проекта TRIK Runtime.

3. Реализация

3.1. Миграция проекта TRIK Runtime

Первое, что сразу хотелось сделать после установки Qt 6 и зависимостей проекта, — сборку TRIK Runtime на новой версии Qt. Грубо говоря, посмотреть, что пойдет не так. Далее нужно было переписать код программы, в котором возникли ошибки, таким образом, чтобы была сохранена существующая логика. Затем необходимо заново провести сборку проекта и повторить процедуру.

В таком итеративном формате были исправлены все ошибки на пути к сборке TRIK Runtime на Qt 6. В предыдущий раз при миграции другого проекта было решено использовать директиву условной компиляции `#ifdef` и макрос `TRIK_USE_QT6`, дабы в зависимости от версии Qt добавлять соответствующий код. Однако в этот раз было принято решение не пользоваться этим, поскольку код проекта сильно разрастется.

Листинг 1: Определение макроса `TRIK_USE_QT6`.

```
!lessThan(QT_MAJOR_VERSION, 6):DEFINES += TRIK_USE_QT6
```

Работу по переписыванию кода можно условно разделить на категории по смыслу.

3.1.1. Переименовывание классов и методов

Большую часть времени заняла данная деятельность. В новой версии Qt часть классов, как и методы в них, были переименованы, удалены, перемещены. Необходимо было в документации фреймворка искать реализацию необходимых классов или методов и использовать новый API. Например, в подпроекте `trikScriptRunner` изначально использовался скриптовый движок, который был реализован в классе `QScriptEngine`¹. В Qt 6 он был полностью удален. Его заменой выступил класс `QJSEngine`².

¹<https://doc.qt.io/qt-5/qscriptengine.html>

²<https://doc.qt.io/qt-6/qjsengine.html>

3.1.2. Добавление новой функциональности

В Qt 6 какая-то часть API была удалена, и замены предоставлено не было. В этом случае необходимо было реализовать что-то свое. Например, в Qt 5 была возможность добавлять внутрь скриптового движка написанные на C++ пользовательские функции с определенной сигнатурой с помощью метода `newFunction()`³ и вызывать их изнутри. В Qt 6 такого удобного интерфейса нет, поэтому был реализован класс `userFunctionWrapper`⁴, который выступает оберткой для пользовательских функций, которые хочется добавить в движок.

3.1.3. Разрешение ownership конфликтов

Как было упомянуто выше, в новой версии проекта используется скриптовый движок `QJSEngine`. В процессе миграции `TRIK Runtime` был замечен следующий факт: при вызове деструкторов некоторых объектов происходило обращение к нулевому указателю. Данные объекты в той или иной мере использовались движком. Он имеет следующую особенность реализации:

When data is transferred from C++ to QML, the ownership of the data always remains with C++. **The exception to this rule is when a QObject is returned from an explicit C++ method call: in this case, the QML engine assumes ownership of the object**, unless the ownership of the object has explicitly been set to remain with C++ by invoking `QQmlEngine::setObjectOwnership` with `QQmlEngine::CppOwnership` specified.

Additionally, the QML engine respects the normal QObject parent ownership semantics of Qt C++ objects, and will never delete a QObject instance which has a parent.

Поэтому во время вызова деструктора со стороны C++ объект мог быть уже удален. Нужно было отследить все ситуации, когда такое происходит, и вручную присвоить владение объектом C++, как описано в

³<https://doc.qt.io/qt-5/qscriptengine.html#newFunction>

⁴<https://github.com/l-golofastov/trikRuntime/blob/qt6-version/trikScriptRunner/src/userFunctionWrapper.cpp>

документации. Более того, необходимо было реализовать свою версию метода `evaluate()`⁵, внутри которого проверяется факт передачи владения объектом движку.

Изменённый код доступен в репозитории на GИTНUB [6]. Также был создан `pull request`⁶, дабы зафиксировать изменения.

3.2. Автоматическая сборка TRIK Runtime

После рефакторинга исходного кода остается добавить GИTНUB action для автоматической сборки на Qt 6, чтобы проверять, что вносимые далее изменения не будут нарушать совместимость с данной версией. Это и было сделано.

⁵<https://doc.qt.io/qt-6/qjsengine.html#evaluate>

⁶<https://github.com/l-golofastov/trikRuntime/pull/1>

Заключение

В рамках данной учебной практики были выполнены следующие задачи:

- Исходный код проекта TRIK Runtime был адаптирован под Qt 6, с сохранением существующей логики и успешным прохождением имеющихся юнит-тестов.
- Была добавлена автоматическая сборка на Qt 6 для проекта TRIK Runtime.

Список литературы

- [1] URL: <https://github.com/trikset>.
- [2] URL: <https://doc.qt.io/qt.html>.
- [3] URL: <https://github.com/features/actions>.
- [4] URL: <https://github.com/trikset/trikRuntime>.
- [5] URL: <https://github.com/trikset/trik-pythonqt>.
- [6] URL: <https://github.com/l-golofastov/trikRuntime>.