

Раздел 2. «Информационно-коммуникационные технологии»

МРНТИ: 20.01.01

Ж.Б. Ибраимов

*Карагандинский технический университет им. Абылкаса Сагинова, Караганда, Казахстан
(Email.ru: ibraimov0409@mail.ru)*

Внедрение цифровых моделей GIS в настольные приложения, разработанные на базе фреймворка «Qt»

В этой статье представлено исчерпывающее руководство по интеграции GIS-моделей в настольные приложения с использованием фреймворка Qt, с акцентом на ArcGIS Runtime SDK, предоставляемый Esri. Обсуждается процесс разработки, особое внимание уделяется созданию проекта с использованием API ArcGIS Runtime SDK в рамках фреймворка Qt. В статье сравниваются три шаблона разработки в Qt (QML с Qt Quick, C++ с Qt Quick и C++ с Qt Widgets) и рекомендуется шаблон C++ с Qt Quick за его универсальность и пригодность для программистов на C++. Кроме того, в статье рассматривается выбор между qMake и CMake в качестве системы сборки приложений Qt, приводится сравнение их плюсов и минусов. В нем сообщается, что выбор зависит от конкретных потребностей проекта и предпочтений команды разработчиков. Статья завершается руководством по выбору базовой карты из ArcGIS Runtime SDK API с подробным описанием распространенных базовых карт и их характеристик. Выбранная базовая карта для проекта выделяется как топографическая карта мира. В целом, статья служит ценным ресурсом для разработчиков, желающих интегрировать функциональность GIS в свои настольные приложения с использованием фреймворка Qt и ArcGIS Runtime SDK, предлагая практическую информацию и пошаговые рекомендации на протяжении всего процесса разработки.

Ключевые слова: фреймворк Qt, модели GIS, ArcGIS Runtime SDK, обработка геопространственных данных, рендеринг карт, пространственный анализ, геокодирование, маршрутизация, Qt Creator, QML (Qt Meta-Object Language), C++ с Qt Quick, qMake и CMake.

Введение

Технология геоинформационных систем (GIS) произвела революцию в том, как мы управляем пространственными данными и анализируем их. Она играет решающую роль в различных областях, включая городское планирование, рациональное природопользование, транспорт и реагирование на чрезвычайные ситуации. В связи с растущим спросом на мощные настольные приложения, способные эффективно обрабатывать GIS-данные, интеграция цифровых GIS-моделей в приложения, разработанные с использованием платформы «Qt», приобрела все большее значение. Qt, платформа для разработки кроссплатформенных приложений, предоставляет разработчикам надежную основу для создания многофункциональных настольных приложений.

Интеграция GIS-моделей в настольные приложения, разработанные на основе фреймворка Qt, обладает рядом преимуществ. Во-первых, это позволяет пользователям беспрепятственно получать доступ к геопространственным данным и манипулировать ими, способствуя эффективному управлению данными и их анализу. Пользователи могут импортировать различные форматы геопространственных данных, визуализировать данные на интерактивных картах, выполнять пространственные запросы и проводить расширенный пространственный анализ.

Во-вторых, интеграция функциональности GIS в приложения на базе Qt позволяет разработчикам создавать удобные для пользователя интерфейсы, которые обеспечивают интуитивно понятный просмотр карт, управление слоями и интерактивные инструменты для исследования данных. Это

Раздел 2. «Информационно-коммуникационные технологии»

предоставляет пользователям знакомую и визуально привлекательную среду для взаимодействия с пространственными данными, делая приложение более доступным и увлекательным.

Кроме того, кроссплатформенная совместимость фреймворка Qt гарантирует, что настольные приложения с поддержкой GIS могут быть развернуты в нескольких операционных системах, включая Windows, macOS и Linux. Это расширяет охват приложения, позволяя пользователям с разных платформ пользоваться его возможностями GIS.

Основная часть

При разработке настольных приложений с использованием платформы фреймворка Qt существует несколько вариантов интеграции GIS моделей: через библиотеки GIS и API (QGIS, GDAL и Esri ArcGIS Runtime SDK для Qt). Однако наиболее эффективным и действенным подходом является использование ArcGIS Runtime SDK, предоставляемого компанией Esri.

Esri ArcGIS Runtime SDK, представляет собой всеобъемлющий и многофункциональный набор инструментов для создания GIS-приложений.

Преимущества использования ArcGIS Runtime SDK с Qt [1]:

1) Всесторонняя функциональность GIS: ArcGIS Runtime SDK предоставляет богатый набор инструментов и API-интерфейсов, которые позволяют разработчикам внедрять расширенные возможности GIS в свои приложения. Он предлагает широкий спектр функциональных возможностей, включая обработку геопространственных данных, рендеринг карт, пространственный анализ, геокодирование и маршрутизацию. Он также включает в себя работу с различными форматами геопространственных данных, выполнение пространственного анализа, визуализацию данных на интерактивных картах и доступ к сервисам ArcGIS.

2) Бесшовная интеграция: ArcGIS Runtime SDK разработан для бесшовной интеграции с фреймворком Qt, обеспечивая совместимость и бесперебойную связь между функциональными возможностями GIS и компонентами приложения Qt. Такая интеграция упрощает процесс разработки и улучшает общий пользовательский опыт.

3) Кроссплатформенная совместимость: как фреймворк Qt, ArcGIS Runtime SDK является кроссплатформенным инструментом, что позволяет разработчикам создавать GIS-приложения, которые могут работать в нескольких операционных системах.

Чтобы приступить к разработке настольных приложений для GIS, крайне важно загрузить правильные версии необходимых инструментов: Qt и ArcGIS Runtime SDK. Использование совместимых версий обеспечивает плавную интеграцию и позволяет избежать проблем с совместимостью между SDK и фреймворком.

Установка необходимых компонентов

В первую очередь следует посетить официальные веб-страницы Qt и Esri (<https://www.qt.io/blog>) [2], чтобы получить самую свежую информацию о рекомендуемых версиях Qt и ArcGIS Runtime SDK для интеграции (таблица 1).

Таблица 1. Совместимые версии Qt и ArcGIS Runtime SDK (на ноябрь 2023 года)

Версия фреймворка Qt	Версия ArcGIS Runtime SDK
$\geq 6.5.1$	200.2
5.12	100.5
5.6.x	100.x

В данной статье были использованы новейшие Qt версия 6.6.0, а также ArcGIS Runtime SDK версия 200.2.0. После успешной загрузки Qt (по ссылке <https://www.qt.io/download>), были выбраны следующие компоненты фреймворка: Qt Creator 11.0.3, Qt 6.6.0 MSVC 2019 64-bit, Qt Designer 6.6.0 MSVC 2019 64-bit, Qt Assistant 6.6.0 MSVC 2019 64-bit (рис. 1).

Раздел 2. «Информационно-коммуникационные технологии»

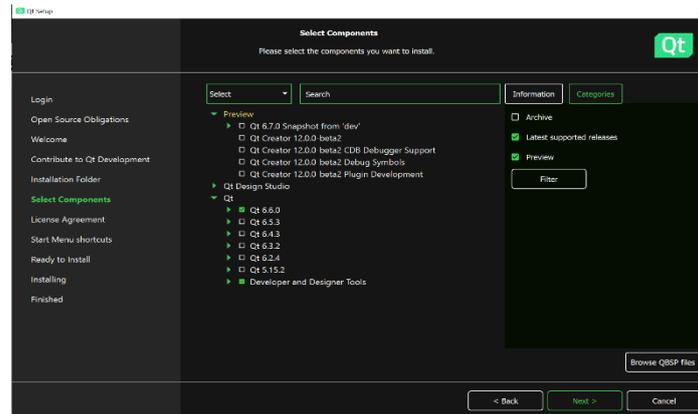


Рисунок 1. Установка фреймворка Qt и его компонентов

Для загрузки ArcGIS Runtime SDK на необходимую операционную систему (в данной статье на Windows), необходимо посетить их официальную веб-страницу, зарегистрировать персональный аккаунт и перейти в раздел загрузки (<https://developers.arcgis.com/downloads/>) (рис. 2). Также здесь можно получить персональный API-ключ, для пользования возможностями SDK.

ArcGIS Maps SDK for Qt

v200.2.0 - August 16, 2023 - [Release notes](#)

Type	Size	SHA256 checksum	
Windows	315 MB	02136880f6399ae821407df3ebf5495d4651eb0c68ae25d539682a4e924e74c6	ArcGIS_Maps_SDK_Qt_Windows_200_2_0.exe
Linux (64 bit)	306.4 MB	69592926c08f083f8436249d2edc3b117c061370f8e7cbcb95c0b66e77832bec	ArcGIS_Maps_SDK_Qt_Linux64_200_2_0.tar.gz
macOS	576.1 MB	6a1b9b355c8cd896467f32159400133d8459dc3d1c1175e5279df2657fa51a21	ArcGIS_Maps_SDK_Qt_macOS_200_2_0.zip
Documentation	271 MB	522ee539a242326e229dc3bfe35127dd3c2a4ef304edd992e47ee9f165d58cef	arcgis-maps-sdk-qt-documentation-200-2-0.zip

Рисунок 2. Загрузка последней версии ArcGIS Runtime SDK для фреймворка Qt

Как упоминалось ранее в преимуществах ArcGIS Runtime SDK, данный API имеет бесшовную установку и она проходит всего в 2 этапа: установка главных компонентов (SDK) на компьютер и интеграция API с Qt Creator (рис. 3).

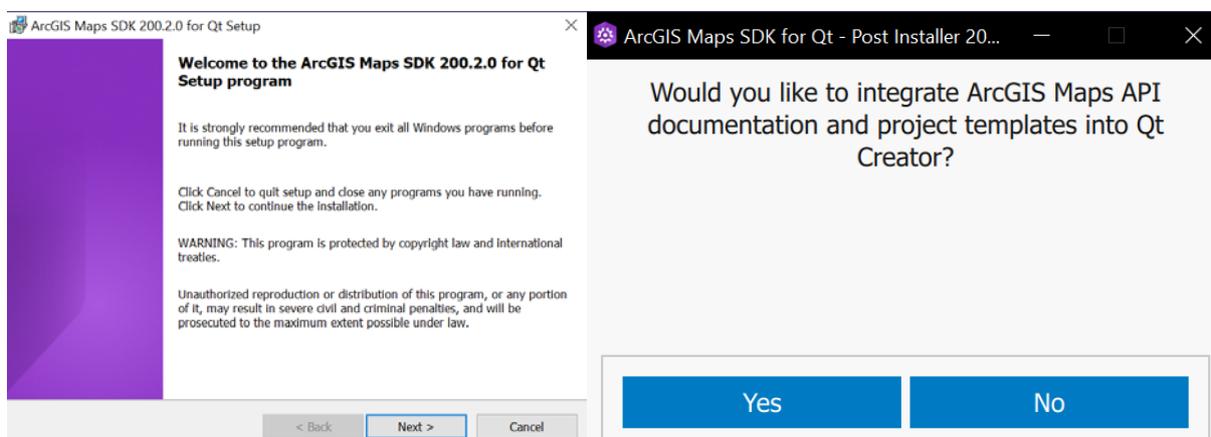


Рисунок 3. Установка главных компонентов ArcGIS Runtime SDK и его интеграция Qt Creator

Раздел 2. «Информационно-коммуникационные технологии»

Создание проекта

Создание проекта на Qt с использованием API ArcGIS Runtime SDK проходит в несколько стадий. Существует три паттерна разработки приложений на Qt: разработка с QML с помощью Qt Quick, разработка с C++ с помощью Qt Quick, а также разработка с C++ с использованием Qt Widgets. Отличие трех паттернов друг от друга представлено в таблице 2.

Таблица 2. Отличие трех паттернов разработки приложений на Qt.

Паттерн разработки	Используемый API	Кому лучше всего подходит паттерн?	Запускается на Linux, macOS, Windows	Запускается на iOS и Android
QML с Qt Quick	QML	Веб-разработчик	Да	Да
C++ с Qt Quick	C++	C++-разработчик	Да	Да
C++ с Qt Widgets	C++	C++-разработчик	Да	Нет

С учетом данных отличий, самым распространенным паттерном разработки является C++ с Qt Quick, так как приложения, созданные с использованием этого паттерна, запускаются практически на любых устройствах и подходят для программистов на C++. В данной статье также был использован этот паттерн разработки.

Кроме выбора паттерна разработки необходимо выбрать систему сборки приложения Qt, которых на данный момент существует два основных варианта: qMake и CMake. Слабые и сильные стороны каждого представлены в таблице 3.

Таблица 3. Плюсы и минусы систем сборки приложения Qt – qMake и Cmake [3].

Система сборки	Плюсы	Минусы
qMake	<p><i>Тесная интеграция с Qt:</i> qMake — это система сборки по умолчанию для Qt, и она тесно интегрирована с ним. Это может упростить работу с функциями и библиотеками, специфичными для Qt.</p> <p><i>Простой синтаксис:</i> qMake использует простой и понятный синтаксис, который может быть полезен для небольших и средних проектов. Файлы конфигурации, как правило, короче и более удобочитаемы.</p> <p><i>Неявные правила:</i> QMake имеет неявные правила для создания файлов различных типов, что уменьшает необходимость в явных правилах в файле проекта.</p>	<p><i>Ограниченная гибкость:</i> QMake может не хватать некоторых расширенных функций и гибкости по сравнению с системами сборки более общего назначения, такими как CMake. Это может усложнить реализацию сложных проектов с нетрадиционными конструкциями.</p>
CMake	<p><i>Кроссплатформенная поддержка:</i> CMake известен своей превосходной кроссплатформенной поддержкой. Он генерирует</p>	<p><i>Кривая обучения:</i> CMake имеет более крутую кривую обучения по сравнению с QMake, особенно для начинающих.</p>

Раздел 2. «Информационно-коммуникационные технологии»

	<p>файлы сборки для конкретной платформы (например, Makefiles, файлы проектов Visual Studio) и может адаптироваться к различным операционным системам и компиляторам.</p> <p><i>Расширяемость:</i> СMake обладает высокой степенью расширяемости и позволяет пользователям определять пользовательские правила сборки и скрипты.</p> <p><i>Большая экосистема:</i> у СMake большое и активное сообщество, а это значит, что доступно множество ресурсов и сторонних модулей. Это может быть полезно для поиска решений распространенных проблем сборки.</p>	<p>Конфигурационные файлы могут быть более сложными, и для их понимания может потребоваться больше усилий.</p> <p><i>Менее тесная интеграция с Qt:</i> хотя СMake поддерживает проекты Qt, он может быть интегрирован не так тесно, как QMake. Это может привести к дополнительной ручной настройке некоторых специфичных для Qt функций.</p>
--	---	---

Таким образом, выбор между qMake и СMake зависит от конкретных потребностей проекта и предпочтений команды разработчиков. qMake может больше подойти для более простых проектов с акцентом на разработку Qt (в связи с этим в данной статье была использована система сборки qMake), в то время как СMake обеспечивает большую гибкость и кроссплатформенную поддержку для более крупных и сложных проектов.

Следующим шагом создания проекта является выбор базовой карты API ArcGIS Runtime SDK [4]. ArcGIS предоставляет множество базовых карт, которые пользователи могут использовать в картографических приложениях. Esri, компания, стоящая за ArcGIS, периодически обновляет и добавляет новые базовые карты, поэтому вполне возможно, что в будущем могут появиться дополнительные базовые карты или изменения. Однако некоторые из распространенных базовых карт, доступных в ArcGIS, вместе с их основными характеристиками представлены в таблице 4.

Таблица 4. Базовые карты API ArcGIS Runtime SDK.

Название базовой карты	Описание
World Imagery	Спутниковые и аэрофотоснимки высокого разрешения, охватывающие весь земной шар.
World Street Map	Подробная базовая карта с информацией об уровне улиц, надписями и нейтральной цветовой гаммой.
World Topographic Map	Топографическая карта с контурными линиями, транспортными сетями и растительным покровом.
World Terrain Base	Базовая карта местности с заштрихованным рельефом, тонировкой высот и контурными линиями.
World Light Gray Base	Базовая карта светлого цвета с минимальным количеством надписей и элементов.

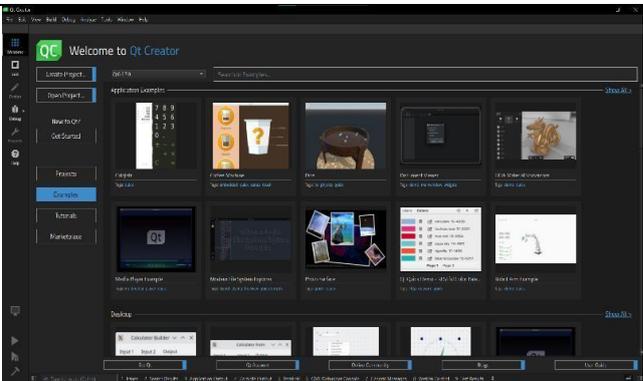
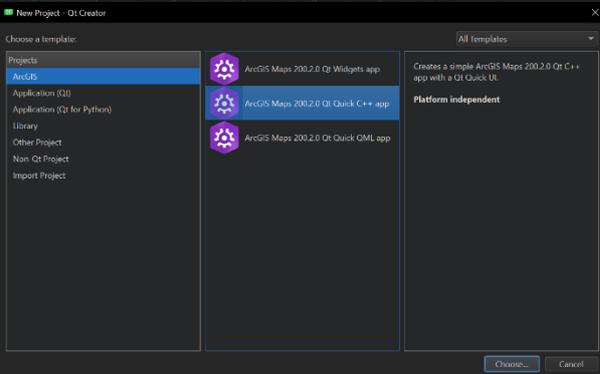
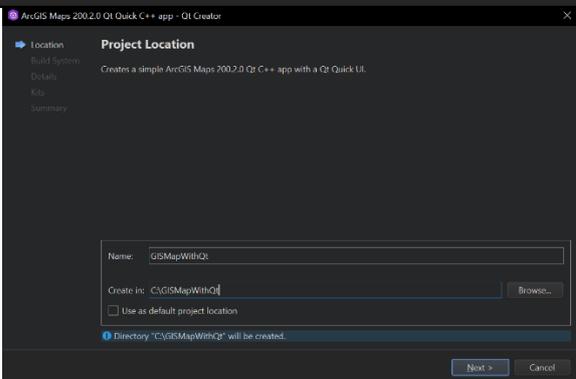
Раздел 2. «Информационно-коммуникационные технологии»

World Oceans Basemap	Базовая карта, ориентированная на океаны и морские объекты.
----------------------	---

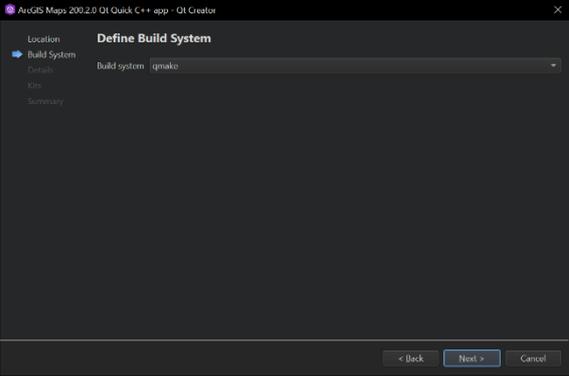
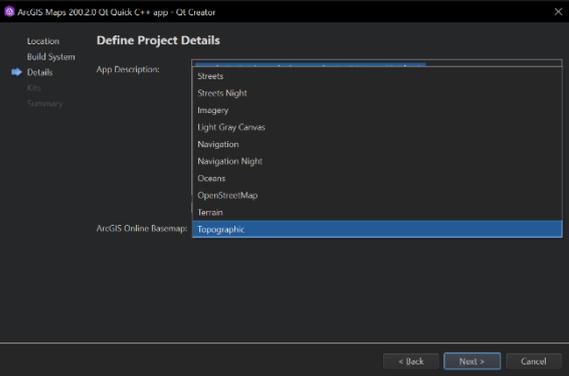
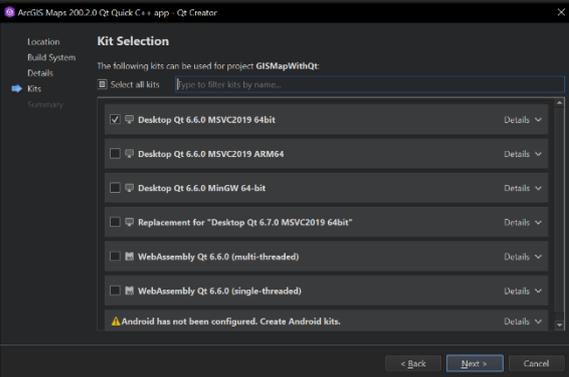
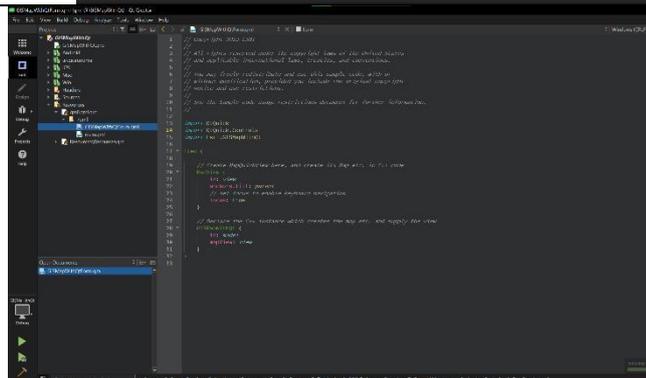
При работе с ArcGIS пользователи могут выбрать наиболее подходящую базовую карту для своих конкретных картографических потребностей, учитывая такие факторы, как назначение карты и тип представляемых географических данных. Всегда полезно проверить онлайн-платформу ArcGIS на наличие последних предложений по базовым картам (<https://blogs.esri-cis.com/2019/07/05/basemap/>). Также отдельного упоминания стоит то, что ArcGIS предоставляет некоторые базовые карты в 3D-формате. В данной статье для проекта была выбрана World Topographic Map в качестве базовой карты.

Стадии создания среды разработки приложения Qt с использованием API ArcGIS Runtime SDK представлены в таблице 4.

Таблица 4. Стадии создания среды разработки.

Стадия	Описание
	Главный экран <i>Qt Creator</i> , который появляется перед разработчиками после открытия приложения. Чтобы перейти на следующую стадию необходимо нажать на кнопку «Create Project...».
	Окно выбора паттерна разработки: ArcGIS Maps 200.2.0 Qt Widgets app ArcGIS Maps 200.2.0 Qt Quick C++ app ArcGIS Maps 200.2.0 Qt Quick QML app Чтобы перейти на следующую стадию необходимо выбрать ArcGIS Maps 200.2.0 Qt Quick C++ app.
	Окно выбора местоположения проекта и названия. Можно выбрать произвольное место и название. Для данного проекта было выбрано название <i>GISMapWithQt</i> .

Раздел 2. «Информационно-коммуникационные технологии»

	<p>Окно выбора системы сборки: qMake или CMake. Чтобы перейти на следующую стадию необходимо выбрать систему сборки qMake.</p>
	<p>Окно выбора базовой карты и добавления описания проекта Описанием можно выбрать произвольное. Чтобы перейти на следующую стадию необходимо выбрать базовую карту Topographic.</p>
	<p>Окно выбора инструмента разработки (Development Kit) Так как во время загрузки Qt была выбрана компонента Qt 6.6.0 MSVC 2019 64-bit, чтобы перейти на следующую стадию необходимо выбрать её в списке.</p>
	<p>Экран созданного проекта Qt.</p>

Создание точки на карте

Для добавления точки на Topographic карте, необходимо в папке «Headers» проекта открыть файл <GISMapWithQt>.h (на месте <> будет выбранное название проекта) (рис. 4).

Раздел 2. «Информационно-коммуникационные технологии»

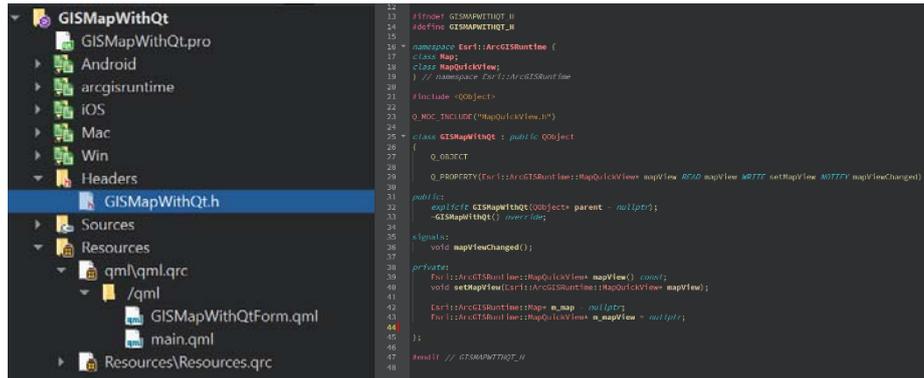


Рисунок 4. Главный Header-файл и его содержимое.

В поле *private* в классе <GISMapWithQt> необходимо добавить новый метод *setPointOnMap* (название метода может быть произвольным) (рис. 5).



Рисунок 5. Объявление метода *setPointOnMap* в классе <GISMapWithQt>. Далее необходимо открыть файл <GISMapWithQt>.cpp в папке Sources проекта (рис. 6).

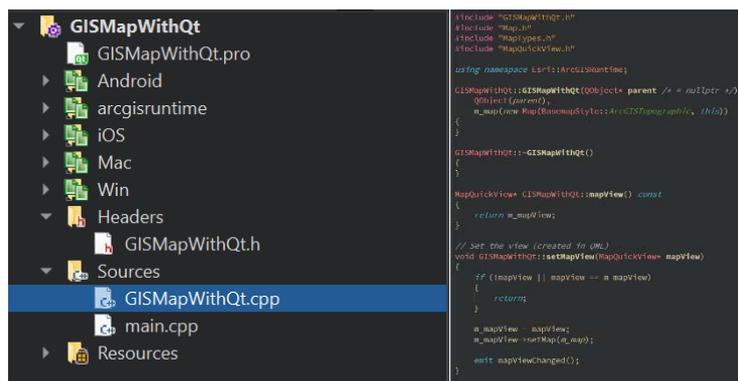


Рисунок 6. Файл <GISMapWithQt>.cpp и его содержимое.

В данном файле необходимо добавить использование следующих header-файлов: *Point.h*, *Viewpoint.h*, *SpatialReference.h*, *TaskWatcher.h*, а также добавить определение метода *setPointOnMap* (рис. 7).

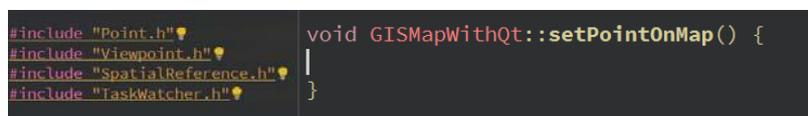


Рисунок 7. Добавление header-файлов и определение метода *setPointOnMap*.

Раздел 2. «Информационно-коммуникационные технологии»

В тело метода *setPointOnMap* необходимо добавить следующие строки кода (рис. 8):

1) `const Point center(72.983194, 50.064645, SpatialReference::wgs84());`

Эта строка создает объект `Point` [5] с именем `center`. `Point` представляет географическую точку с широтой, долготой и, необязательно, высотой. В этом случае широта равна 50.064645, долгота равна 72.983194 (можно выбрать произвольную точку), а пространственная привязка установлена на `SpatialReference::wgs84()` [6], что указывает на систему координат WGS84.

2) `const Viewpoint viewpoint(center, 100000.0);`

Эта строка создает объект `Viewpoint` [7] с именем `viewpoint`. Точка обзора в ArcGIS представляет собой определенный вид карты, включая центральную точку и масштаб. Точка обзора инициализируется ранее созданной центральной точкой и масштабом 100 000,0.

3) `m_mapView->setViewpoint(viewpoint);`

Эта строка устанавливает точку обзора объекта `MapView` (`m_mapView`) на указанную точку обзора. Функция `setViewpoint` используется для определения начального вида карты при ее отображении.

```
void GISMapWithQt::setPointOnMap() {
    const Point center(72.983194, 50.064645, SpatialReference::wgs84());
    const Viewpoint viewpoint(center, 100000.0);
    m_mapView->setViewpoint(viewpoint);
}
```

Рисунок 8. Тело метода *setPointOnMap*.

В этом же файле необходимо добавить вызов метода *setPointOnMap* в автоматический сгенерированный метод *setMapView* класса `<GISMapWithQt>` (рис. 9).

```
void GISMapWithQt::setMapView(MapQuickView* mapView)
{
    if (!mapView || mapView == m_mapView)
    {
        return;
    }

    m_mapView = mapView;
    m_mapView->setMap(m_map);
    setPointOnMap();
    emit mapViewChanged();
}
```

Рисунок 9. Вызов метода *setPointOnMap* в методе *setMapView*.

Следующим этапом является добавление API-ключа в файл *main.cpp* проекта, а именно в переменную `const apiKey` (рис. 10).

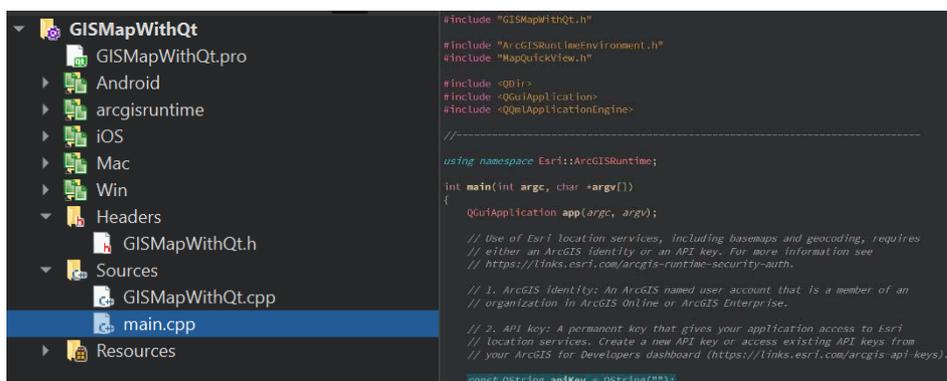


Рисунок 10. Файл *main.cpp*, а также переменная `apiKey`.

Раздел 2. «Информационно-коммуникационные технологии»

После успешной сборки проекта и запуска приложения перед пользователем будет открыта карта, выбранной местности (рис. 11).

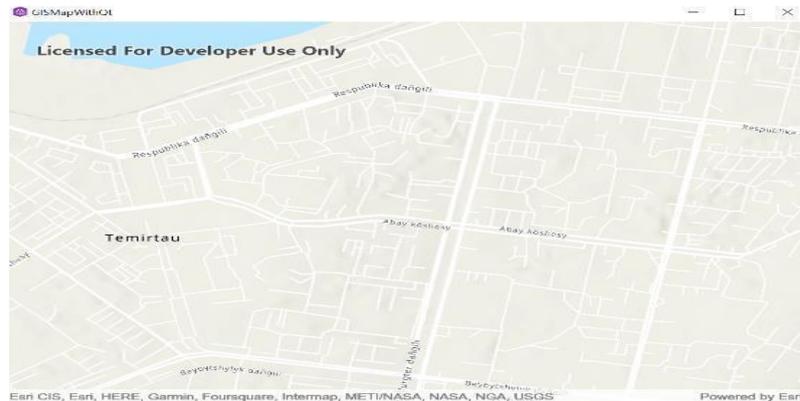


Рисунок 11. Финальный результат.

Вывод

В заключение, эта статья служит исчерпывающим руководством для разработчиков, желающих интегрировать функциональность географической информационной системы (GIS) в свои настольные приложения с использованием фреймворка Qt и ArcGIS Runtime SDK, предоставляемых Esri. Подчеркиваются преимущества выбора ArcGIS Runtime SDK для Qt, такие как его надежная функциональность ГИС, плавная интеграция с фреймворком Qt и кроссплатформенная совместимость.

В статье подробно рассказывается о процессе установки необходимых компонентов, подчеркивается важность использования совместимых версий Qt и ArcGIS Runtime SDK для обеспечения бесперебойной интеграции. Предоставляя практические шаги, включая загрузку и установку инструментов, статья способствует четкому пониманию начальной настройки. Процесс разработки подробно изучается с акцентом на выборе подходящего шаблона разработки, выборе системы сборки (qMake или CMake) и создании проекта с использованием ArcGIS Runtime SDK API в рамках Qt framework. Сравнение различных моделей разработки и систем сборки дает разработчикам ценную информацию для принятия обоснованных решений, основанных на их требованиях к проекту и предпочтениях команды. Кроме того, в статье рассматривается важнейший аспект выбора базовой карты с помощью ArcGIS Runtime SDK API, предлагается подробный обзор распространенных базовых карт и их характеристик. Выбрав топографическую карту мира в качестве базовой карты для проекта, в статье приводится практический пример, который поможет разработчикам в процессе принятия решений.

Список использованных источников

1. Официальная страница ArcGIS Runtime SDK: <https://developers.arcgis.com/qt/cpp/api-reference/>.
2. Официальная страница фреймворка Qt: <https://www.qt.io/blog>.
3. <https://forum.qt.io/topic/117660/cmake-vs-qmake-which-is-better-to-use-for-new-project>
4. Официальная страница про базовые карты: <https://www.esri.com/en-us/arcgis/products/arcgis-platform/services/basemaps>.
5. Класс Point: <https://developers.arcgis.com/qt/cpp/api-reference/esri-arcgisruntime-point.html>.
6. Класс SpatialReference: <https://developers.arcgis.com/qt/cpp/api-reference/esri-arcgisruntime-spatialreference.html>.
7. Класс Viewpoint: <https://developers.arcgis.com/qt/cpp/api-reference/esri-arcgisruntime-viewpoint.html>.

Раздел 2. «Информационно-коммуникационные технологии»

Ж.Б. Ибраимов

«Qt» фреймворкінің негізінде әзірленген жұмыс үстелі қосымшаларына GIS цифрлық модельдерін енгізу

Бұл мақалада ESRI ұсынған ArcGIS Runtime SDK-ге баса назар аударып, qt шеңберін қолдана отырып, GIS модельдерін жұмыс үстелі қосымшаларына біріктіру туралы толық нұсқаулық берілген. Даму процесі талқыланады, Qt шеңберіндегі ArcGIS Runtime SDK API көмегімен жобаны құруға ерекше назар аударылады. Мақалада Qt-дегі үш даму үлгісі салыстырылады (QML Qt Quick-пен, C Q Qt Quick-пен және C и Qt виджеттерімен) және C қи-мен Qt Quick-пен оның әмбебаптығы мен C ПРОГРАММ бағдарламашыларына жарамдылығы үшін ұсынылады. Сонымен қатар, мақалада qt қосымшаларын құрастыру жүйесі ретінде qMake және CMake арасындағы таңдау қарастырылады, олардың артықшылықтары мен кемшіліктерін салыстыру келтірілген. Бұл таңдау жобаның нақты қажеттіліктеріне және әзірлеушілер тобының қалауына байланысты екенін хабарлайды. Мақала ArcGIS Runtime SDK API-ден базалық картаны таңдау бойынша нұсқаулықпен аяқталады, жалпы базалық карталар мен олардың сипаттамалары егжей-тегжейлі сипатталған. Жоба үшін таңдалған негізгі карта әлемнің топографиялық картасы ретінде ерекшеленеді. Тұтастай алғанда, мақала qt және ArcGIS Runtime SDK шеңберін қолдана отырып, GIS функционалдығын жұмыс үстелі қосымшаларына біріктіргісі келетін әзірлеушілер үшін құнды ресурс ретінде қызмет етеді, бүкіл даму процесінде практикалық ақпарат пен қадамдық нұсқаулар ұсынады.

Түйін сөздер: Qt шеңбері, GIS модельдері, ArcGIS Runtime SDK, геокеңістіктік деректерді өңдеу, карталарды көрсету, кеңістіктік талдау, геокодтау, маршруттау, Qt Creator, QML (Qt Meta-Object Language), C Q Qt Quick, qMake және CMake көмегімен.

J.B. Ibrahimov

Implementation of digital GIS models in desktop applications developed on the basis of the «Qt» framework

This article provides a comprehensive guide to integrating GIS models into desktop applications using the Qt framework, with an emphasis on the ArcGIS Runtime SDK provided by Esri. The development process is discussed, special attention is paid to creating a project using the ArcGIS Runtime SDK API within the framework of the Qt framework. The article compares three development templates in Qt (QML with Qt Quick, C++ with Qt Quick and C++ with Qt Widgets) and recommends the C++ template with Qt Quick for its versatility and suitability for C++ programmers. In addition, the article discusses the choice between qMake and CMake as a Qt application build system, and compares their pros and cons. It states that the choice depends on the specific needs of the project and the preferences of the development team. The article concludes with a guide on choosing a base map from the ArcGIS Runtime SDK API with a detailed description of common base maps and their characteristics. The selected base map for the project stands out as a topographic map of the world. In general, the article serves as a valuable resource for developers who want to integrate GIS functionality into their desktop applications using the Qt framework and the ArcGIS Runtime SDK, offering practical information and step-by-step recommendations throughout the development process.

Раздел 2. «Информационно-коммуникационные технологии»

Keywords: Qt framework, GIS models, ArcGIS Runtime SDK, geospatial data processing, map rendering, spatial analysis, geocoding, routing, Qt Creator, QML (Qt Meta-Object Language), C++ with Qt Quick, qMake and CMake.

References

1. Official page of the ArcGIS Runtime SDK: <https://developers.arcgis.com/qt/cpp/api-reference/>.
2. Official page of the Qt framework: <https://www.qt.io/blog>.
3. <https://forum.qt.io/topic/117660/cmake-vs-qmake-which-is-better-to-use-for-new-project>
4. The official page about the base maps: <https://www.esri.com/en-us/arcgis/products/arcgis-platform/services/basemaps>.
5. Point Class: <https://developers.arcgis.com/qt/cpp/api-reference/esri-arcgisruntime-point.html>.
6. SpatialReference class: <https://developers.arcgis.com/qt/cpp/api-reference/esri-arcgisruntime-spatialreference.html>.
7. Viewpoint class: <https://developers.arcgis.com/qt/cpp/api-reference/esri-arcgisruntime-viewpoint.html>.