

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б10-мм

Карасева Елизавета Олеговна

Процесс сопровождения переводов в графической оболочке контроллера TRIK

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
старший преподаватель кафедры системного программирования, Я. А. Кириленко

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Локализация с помощью инструментов от Qt	6
2.1.1. Qt Linguist	7
2.1.2. lupdate	8
2.1.3. lrelease	9
2.1.4. Комментарии и дополнения	9
2.2. Веб-клиенты для локализации приложений	10
2.3. Используемые технологии	11
3. Метод	13
3.1. Основные проблемы	13
3.1.1. Иерерхия файлов	13
3.1.2. Взаимодействие с консолью	16
3.2. Документация	16
4. Тестирование и апробация	18
Заключение	19
Список литературы	20

Введение

Контроллер TRIK является частью проекта TRIK, который занимается разработкой программного обеспечения для обучения робототехнике. Графическая оболочка контроллера TRIK написана на языке программирования C++ с использованием библиотеки Qt, и отвечает за обработку входных данных (например, с датчиков), связь с аппаратными компонентами робота, а также выполнение команд на машинах.

На данный момент интерфейс графической оболочки представлен на английском и русском языках: в основном этим продуктом пользуются в России под операционной системой Windows. Однако с выходом на глобальный рынок возникает потребность адаптации интерфейса к различным языкам: контроллер используется в работе с младшими школьниками, и в данной ситуации интерфейс на иностранном языке может вызвать непонимание детей.

По информации от коллектива разработчиков TRIK, предложения по адаптации проекта на разные языки (бразильский, французский) поступали от нетехнических специалистов не один раз. К тому же, в России есть интерес по адаптации на национальные языки (татарский, башкирский) в рамках программ поддержки национальных традиций.

Решение этой проблемы — локализация приложения. В широком смысле это адаптация интерфейса для определённой страны или региона [24] с целью расширения аудитории продукта. Перевод текста на другой язык — лишь часть действий по локализации: помимо этого нужно понимать контекст использования слов и формата чисел, изменять интерфейс в соответствии, например, с длиной получившихся строк и, самое главное, процесс добавления переводов в приложение должен быть удобным и понятным для нетехнических специалистов.

В целом, компоненты TRIK готовы к поддержке переводов. Однако, в связи с архитектурными решениями, принятыми разработчиками, на данный момент этот процесс организован не в соответствии с официальной документацией Qt, что добавляет технических трудностей переводчикам.

Поэтому в рамках учебной практики необходимо было реализовать процесс, который будет удобен и для разработчиков, и для нетехнических специалистов.

1. Постановка задачи

Целью работы является модернизация процесса добавления переводов в графическую оболочку контроллера TRIK в соответствии со стандартами Qt. Для её выполнения были поставлены следующие задачи.

1. Провести обзор средств для локализации Qt-приложений:
 - Утилит, предоставляемых Qt;
 - Веб-клиентов поверх Qt Linguist;
2. Провести реинжиниринг процесса в соответствии со стандартами Qt, а именно:
 - Упростить иерархию построения файлов перевода;
 - Избавиться от прямого взаимодействия с консолью из исходного кода приложения;
3. Протестировать работоспособность кода в trikRuntime;
4. Составить инструкцию по добавлению переводов для нетехнических специалистов.

В случае успешного внедрения процесса переводов будет проведен реинжиниринг других компонент проекта TRIK, в том числе и TRIK Studio, где текстовых строк в несколько десятков раз больше.

2. Обзор

Рассмотрим, как организовать процесс создания переводов для Qt-приложений, какие инструменты и веб-клиенты можно для этого использовать, поймем, как на данный момент происходит взаимодействие с переводами в проекте TRIK на примере графической оболочки контроллера TRIK и объясним выбор использованных технологий в рамках данной учебной практики.

2.1. Локализация с помощью инструментов от Qt

Итак, чтобы перевести приложение на другой язык, предусмотрен следующий порядок действий [13]:

1. В исходном коде приложения строки, подлежащие переводу, маркируются макросами `tr()` или `QObject::tr()` [20]. Для работы с такими строками необходимо использовать класс `QString`, обладающий множеством удобных функций для работы с текстовыми данными в Qt-приложениях.
2. После того, как все строки отмечены, вызывается инструмент `lupdate`. Он извлекает переводимый текст из исходного кода приложения на C++ и создает файлы TS (Translation Source) для переводчиков.
3. Сгенерированные TS файлы открываются с помощью сторонних инструментов для работы с переводами: Qt Linguist-a [14] или каких-либо веб-клиентов. Так как файлы TS представляются в формате XML, их можно редактировать вручную, добавляя фразы на нужном языке для каждой строки.
4. После этого нужно воспользоваться инструментом `lrelease`, который компилирует файлы TS в бинарные файлы перевода QM (Qt message), которые, в свою очередь, будут использованы для отображения соответствующего языка интерфейса во время выполнения приложения.

5. И наконец, в исходный код приложения добавляется поддержка выбора языка с использованием класса `QTranslator` [11] и других инструментов Qt для работы с локалью [10].

По сути, файлы TS можно назвать «исходными файлами», а файлы QM — «объектными файлами». Работа переводчика заключается в редактировании файлов TS, но пользователям приложения нужны только файлы QM. Оба типа файлов не зависят от платформы и локали.

Листинг 1: Пример TS файла с переводом

```
<?xml version="1.0" encoding="UTF-8"?>
<TS version="2.0" language="en-GB">
  <context>
    <message>
      <source>boolean_key</source>
      <translation>— true</translation>
    </message>
    <message>
      <source>unverified_key</source>
      <translation>This translation is not yet verified and waits
        for it</translation>
    </message>
  </context>
</TS>
```

Далее рассмотрим инструменты для локализации, предоставляемые библиотекой Qt.

2.1.1. Qt Linguist

Ключевым инструментом для локализации Qt-приложений, как было сказано выше, является Qt Linguist. Одним из главных его достоинств является удобный интерфейс: пользователи могут просматривать и редактировать строки, предоставляя контекст и подсказки для луч-

шего понимания использования слов. Также можно менять статус добавленных строк: незакончена, завершена, устарела и т.д.

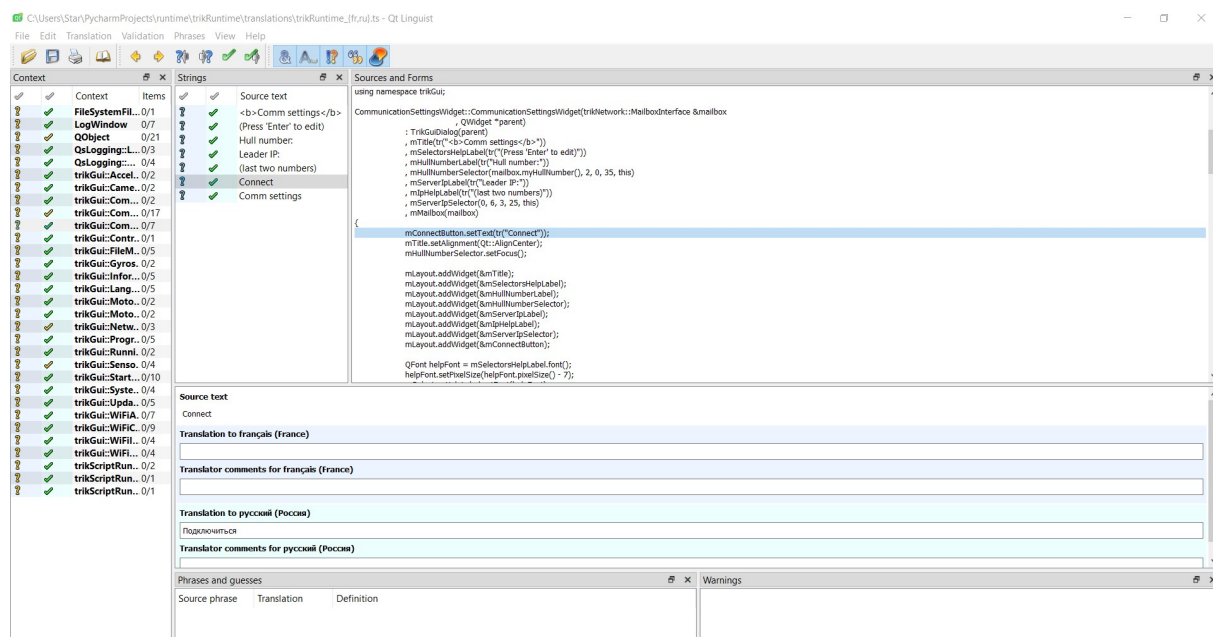


Рис. 1: Интерфейс программы Qt Linguist

Также с помощью Qt Linguist можно управлять переводами сразу на несколько языков в рамках одного проекта, что особенно важно для многоязычных приложений, и к тому же позволяет нескольким переводчикам совместно работать над одним проектом.

2.1.2. lupdate

Эта утилита автоматически обнаруживает конструкторы `tr()` и `QT_TR*_NOOP()` — то есть строки, которые нуждаются в переводе — и создает или обновляет для них файлы TS (обычно один на каждый язык). Эту команду можно вызывать с разными опциями: только на основной файл сборки проекта или с указанием конкретных TS файлов; генерировать перевод для определенного языка или игнорировать строки с определенными параметрами.

Листинг 2: Пример вызова lupdate из командной строки

```
> lupdate trikRuntime.pro
Updating 'translations/trikRuntime_ru.ts'...
```

```
Found 149 source text(s) (149 new and 0 already existing)
Updating 'translations/trikRuntime_fr.ts'...
Found 149 source text(s) (149 new and 0 already existing)
```

где `trikRuntime.pro` — основной файл проекта.

2.1.3. lrelease

Эта утилита от Qt используется для компиляции файлов перевода (.ts) в бинарные файлы (.qm), которые содержат эффективные структуры данных для поддержки быстрого поиска и использования переводов во время выполнения приложения.

Листинг 3: Пример вызова функции lrelease

```
> lrelease trikRuntime.pro
Updating 'translations/trikRuntime_ru.qm'...
Generated 131 translation(s) (125 finished and 6 unfinished)
Ignored 18 untranslated source text(s)
Updating 'translations/trikRuntime_fr.qm'...
Generated 0 translation(s) (0 finished and 0 unfinished)
Ignored 149 untranslated source text(s)
```

где `trikRuntime.pro` — основной файл проекта.

После генерации бинарные файлы перевода используются в Qt-приложении для отображения текста на различных языках в зависимости от текущей локали пользователя.

2.1.4. Комментарии и дополнения

Взаимодействие с `lupdate` и `lrelease` можно организовать через командную строку, или же прописать необходимые команды в файлах сборки приложения (.pro).

На данный момент в графической оболочке контроллера TRIK не предложено никакого прямого взаимодействия с переводами. Однако сам `trikRuntime` предварительно готов к этому — все строки записаны с использованием макроса `tr()`, также используются `lupdate` и

`lrelease`, но это сделано нестандартно: утилиты вызываются из кода напрямую через консольные команды (подробнее в разделе с реализацией).

2.2. Веб-клиенты для локализации приложений

Часто во время создания переводов приложения поверх Qt Linguist используют какой-либо веб-клиент — это облегчает организацию совместной работы переводчиков, а также позволяет работать с файлами из любого места, где есть доступ к интернету.

С использованием ChatGPT [22], Google и анализа проектов на Github [1] был проведен обзор инструментов для локализации Qt приложений и найдены подходящие варианты. При анализе инструментов учитывались следующие критерии: популярность инструмента (количество проектов, в которых он применяется), доступность по цене и ограничения на бесплатные версии. При этом не учитывались инструменты, не поддерживающие взаимодействие с TS файлами.

Полный обзор сравнения по параметрам приведен в таблице 1. Данные актуальны на 14.12.2023.

Благодаря проведенному анализу выяснилось, что:

- полностью бесплатные фреймворки (Zanata [21], Pootle [8]) не обновляются, начиная с 2019 года, так что было решено не рассматривать их далее;
- Lokalise [6] не поддерживает основные теги TS файлов [15], так что его использование в данном проекте не представляется возможным;
- интеграция с Github присутствует во всех инструментах, однако в разных тарифных планах присутствуют ограничения на количество загружаемых проектов, созданных веток, контрибьюторов и пользователей проекта, к тому же Transifex [18] имеет завышенную стоимость тарифных планов, а Localazy [2] — слишком новый инструмент, что не вызывает доверия.

	Основные недостатки	Лицензия	Цена руб/мес	Ограничения плана	Проекты
Pootle [8]	Нет обновлений с 2019 года, требуется python2	GPL	—	—	?
Crowdin [7]	Сложно управлять большим объемом строк	Проприетарное	от \$50	Участники проекта, ветки	174K+
Transifex [18]	Стоимость планов, сложность начальной настройки	Проприетарное	от \$90	Проекты, ветки	40K+
Zanata [21]	Нет обновлений с 2018 года	LGPL	—	—	?
Weblate [19]	—	GPL	от \$20	Языки, строки	2,5K+
Lokalise [6]	Не поддерживает основные теги TS файлов	Проприетарное	от \$120	Проекты	?
Localazy [2]	Относительно новый и неизвестный сервер	Apache 2.0	от \$16	Строки для перевода	?

Таблица 1. Сравнение веб-инструментов для локализации

Резюмируя, можно сказать, что большинство инструментов не стоит рассматривать, однако в данной практике был выбран такой способ добавления переводов, что после его внедрения можно будет использовать поверх Qt Linguist какой-либо из рассмотренных выше веб-инструментов; скорее всего, Crowdin или Transifex. Это еще больше повысит удобство добавления переводов для нетехнических специалистов.

2.3. Используемые технологии

Так как все вышеописанные технологии используются поверх Qt Linguist-a, для создания и редактирования переводов было решено использовать непосредственно Qt Linguist, так как он автоматически предоставляется Qt, не требует дополнительной настройки и оплаты,

и его использование дает возможность последующей интеграции с веб-инструментами.

Для обновления существующих файлов перевода и генерации новых использовались утилиты `lupdate` и `lrelease`, также предоставляемые Qt.

Сборка и взаимодействие с кодом графической оболочки контроллера TRIK проводилась в Qt Creator-е, так как он оптимизирован для работы с Qt [12] (в отличие от своего аналога KDevelop [5]), а проект TRIK базируется на этом фреймворке и является кроссплатформенным. Так что еще одним достоинством Qt Creator-а является поддержка разработки приложений на разных платформах.

Также на одном из этапов работы потребовалось написать скрипт на языке Python. Для работы использовалась среда разработки PyCharm [9], потому что она предоставляет богатый набор инструментов для разработки и отладки Python-приложений, а также в нее можно легко установить необходимые расширения, в моем случае, библиотеку для парсинга xml-файлов.

3. Метод

Перед началом разработки необходимо было собрать проект `trikRuntime` и понять, как реализовано взаимодействие с `lupdate` и `lrelease`. В `trikRuntime` существует отдельный подпроект `translations`: в файле `translations.pro` описывается вся работа с переводами.

3.1. Основные проблемы

Работа проходила в несколько этапов: анализ кода, выявление мест, которые можно улучшить, и, собственно, рефакторинг кода.

3.1.1. Иерерхия файлов

Первая проблема заключалась в том, что по стандартам Qt каждому языку должен соответствовать один файл TS. Но, так как `trikRuntime` является модульным проектом, на момент начала практики файлы переводов создавались для каждого подпроекта и объединялись в папки по названию языка. Также в каждой папке находился файл с названием локали (`locale.ini`) (рис. 2).

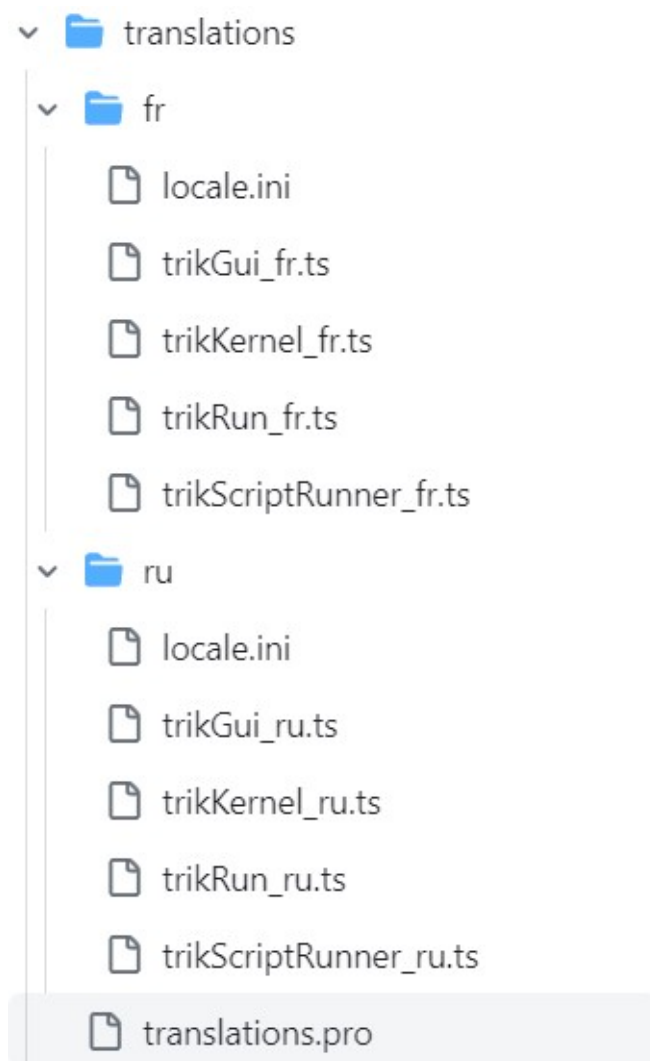


Рис. 2: Иерархия файлов переводов до реинжиниринга

Такая структура затрудняет добавление переводов сторонними пользователями, так как непонятен принцип названия файлов и их связь друг с другом. Также в Qt Linguist нет возможности открывать несколько файлов одновременно: при переводе каждого модуля отдельно можно запутаться в их порядке и числе.

Для повышения удобства взаимодействия с файлами переводов были проведены следующие действия.

1. Собрать файлы переводов в один для каждого языка. Чтобы не переписывать строки вручную, был создан скрипт на языке Python, анализирующий все папки и объединяющий содержащиеся в них TS файлы в один с названием в формате `название_проекта_<код_языка>.ts`.

2. Убрать файлы с названием локали из папок с переводами. Этот файл нужен только для корректного отображения языков в меню выбора языка интерфейса (см. рис. 3). Поэтому было решено модифицировать код приложения таким образом, чтобы получать локаль непосредственно во время выполнения программы по коду языка, содержащегося в названии файла с переводом. Для этого использовались классы `QFileInfo`, `QString` и `QLocale`.

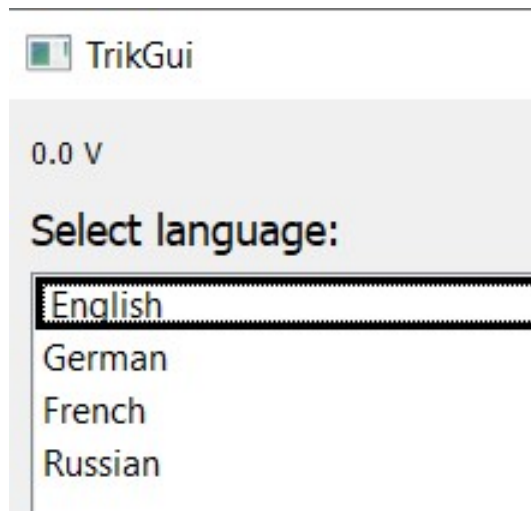


Рис. 3: Меню выбора языка в TrikGui

3. И наконец, нужно было переписать взаимодействие с файлами переводов в соответствии с внесенными изменениями. Для этого использовался класс `QDirIterator`.

После изменений иерархия файлов выглядит следующим образом:

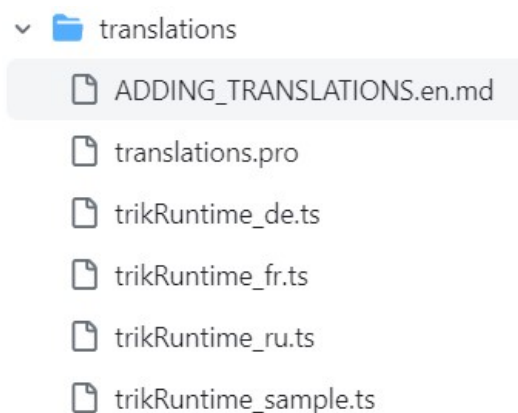


Рис. 4: Файлы перевода после реинжиниринга

3.1.2. Взаимодействие с консолью

Следующая задача заключалась в том, что команды `lrelease` и `lupdate`, вызываясь из кода, напрямую взаимодействовали с консолью (см. листинг 4), хотя по стандартам Qt стоит использовать их как отдельные утилиты.

Листинг 4: Вызов `lupdate` и `lrelease` из файла сборки

```
system(bash -c system_quote(find shell_quote(PWD) -name '*.ts'
-print0 | xargs -0 [QT_HOST_BINS/get]/lrelease --removeidentical))
system(bash -c system_quote(rsync -vrdmR --remove-source-files
--include='*/' --include='*.qm' --exclude='*' ./ TRANSLATIONS_DIR))
system(bash -c system_quote(rsync -vrdmR
--include='*/' --include
='*.ini' --exclude='*' ./ TRANSLATIONS_DIR))
```

В данном фрагменте кода происходит поиск всех TS файлов проекта, по ним генерируются файлы QM, после чего они вручную копируются в каталог с переводами. Моей задачей стало переписать данную часть кода так, чтобы:

- не использовать взаимодействие с консолью;
- не переносить файлы вручную.

Проанализировав исходный код инструмента `lrelease`, выяснилось, что для использования его в качестве отдельной утилиты нужно добавить в код строчку `CONFIG += lrelease`, а также объявить к этому моменту переменные `TRANSLATIONS` и `TRANSLATIONS_DIR`, что и было сделано. Благодаря этому при сборке проекта происходит автоматическое обновление файлов, указанных в `TRANSLATIONS`, и генерация по ним файлов QM в указанную папку, без копирования вручную.

3.2. Документация

Помимо изменения процесса работы с переводами необходимо было добавить в проект соответствующую документацию, так как именно

на неё будут ориентироваться переводчики при работе с проектом. Подробный файл с описанием шагов по добавлению перевода на новый язык и редактированию существующих переводов представлен в репозитории проекта в папке `translations` [23].

4. Тестирование и апробация

Пулл-реквест [16] с модернизацией процесса переводов был предложен в репозиторий проекта, на момент 07.01.2023 тесты в CI проходят, то есть новая версия оболочки контроллера TRIK собирается под `linux` и `macOS`.

Для проведения полной апробации процесса добавления переводов необходимо привлекать сторонних специалистов. Однако из-за сжатых сроков было решено провести минимальную апробацию с использованием методики System Usability Scale (SUS) [17] среди студентов. Участникам было предложено добавить перевод на новый язык, пользуясь предоставляемой инструкцией. Шкала SUS помогает оценить субъективные удобства использования системы, оценка по ней находится в диапазоне от 0 до 100 баллов, среднее значение — 68 баллов. Процесс добавления переводов получил среднюю оценку в 84 балла. Однако следует учитывать, что студенты, принявшие участие в апробации, ранее не работали с инструментами для переводов (Qt Linguist), что делает данную оценку не совсем объективной, так как предполагается, что потенциальные пользователи — переводчики умеют работать с инструментами для редактирования переводов.

Заключение

В результате работы над практикой были решены следующие задачи:

1. Проведен обзор средств для локализации Qt-приложений;
2. Разобрана исходная реализация и переписан процесс взаимодействия с файлами перевода в оболочке контроллера TRIK;
3. Работоспособность кода подтверждается как локально, так и тестами в CI;
4. Составлена инструкция для добавления переводов нетехническими специалистами.

В дальнейшем планируется:

1. Провести аналогичный рефакторинг по добавлению и взаимодействию с переводами для TRIK Studio;
2. Провести апробацию добавленного процесса с привлечением профессиональных переводчиков;
3. Добавить генерацию автоматических переводов с использованием сторонних ресурсов (etc. Google Translate [4]).

Реализация находится в Github-репозитории по ссылке [3] в ветке `master`, документация находится по ссылке [23], разработка ведется под аккаунтом `kleo-53`. Также был создан pull request в репозиторий проекта по ссылке [16].

Список литературы

- [1] Code search results. — URL: <https://github.com/search?q=qmake+lupdate&type=code> (online; accessed: 2023-12-15).
- [2] Content Software localization on autopilot. — URL: <https://localazy.com/> (online; accessed: 2023-12-15).
- [3] GitHub - kleo-53/trikRuntime: Runtime for TRIK controller. — URL: <https://github.com/kleo-53/trikRuntime> (online; accessed: 2024-01-07).
- [4] Google Переводчик. — URL: <https://translate.google.com/?hl=ru> (online; accessed: 2024-01-07).
- [5] KDevelop. A cross-platform IDE for C, C++, Python, QML/-JavaScript and PHP. — URL: <https://kdevelop.org/> (online; accessed: 2023-12-20).
- [6] A Localization and Translation Software Tool | Lokalise. — URL: https://lokalise.com/?_gl=1*14etv1u*_ga*MTE50DQwODMxNC4xNzAxMDg4NjIy*_ga_1JRZETG1RY*MTcwMjc0NTcxNy4zLjEuMTcwMjc0NTc5My4wLjAuMA.. (online; accessed: 2023-12-15).
- [7] Localized software in a few clicks with Crowdin. — URL: <https://crowdin.com/> (online; accessed: 2023-12-12).
- [8] Online translation tool | Pootle. — URL: <https://pootle.translatehouse.org/> (online; accessed: 2023-12-14).
- [9] PyCharm: all about the most popular Python IDE. — URL: <https://datascientest.com/en/pycharm-all-about-the-most-popular-python-ide> (online; accessed: 2023-12-20).
- [10] QLocale Class | Qt Core 5.15.16. — URL: <https://doc.qt.io/qt-5/qlocale.html> (online; accessed: 2023-12-15).

- [11] QTranslator Class | Qt Core 6.6.1. — URL: <https://doc.qt.io/qt-6/qtranslator.html> (online; accessed: 2023-12-15).
- [12] Qt | Tools for Each Stage of Software Development Lifecycle. — URL: <https://www.qt.io/> (online; accessed: 2023-12-20).
- [13] Qt 4.7: Интернационализация с Qt | Документация. — URL: <http://doc.crossplatform.ru/qt/4.7.x/internationalization.html> (online; accessed: 2023-12-15).
- [14] Qt Linguist Manual. — URL: <https://doc.qt.io/qt-5/qtlinguist-index.html> (online; accessed: 2023-12-14).
- [15] Qt Linguist (.ts) | Lokalise Docs. — URL: <https://docs.lokalise.com/en/articles/1400804-qt-linguist-ts> (online; accessed: 2023-12-15).
- [16] Rewrite translation process by kleo-53 · Pull Request 745 · trikset/trikRuntime · GitHub. — URL: <https://github.com/trikset/trikRuntime/pull/745> (online; accessed: 2024-01-07).
- [17] System Usability Scale (SUS) | Usability.gov. — URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (online; accessed: 2023-12-20).
- [18] Transifex | The best platform to continuously localize any digital content. — URL: <https://www.transifex.com/> (online; accessed: 2023-12-12).
- [19] Weblate - перевод с помощью сетевых технологий. — URL: <https://weblate.org/ru/> (online; accessed: 2023-12-12).
- [20] Writing Source Code for Translation | Qt 6.6. — URL: <https://doc.qt.io/qt-6/i18n-source-translation.html> (online; accessed: 2023-12-14).
- [21] Zanata: Community powered translation. — URL: <http://zanata.org/> (online; accessed: 2023-12-15).

- [22] chat.openai.com. — URL: <https://chat.openai.com/> (online; accessed: 2023-12-15).
- [23] trikRuntime/translations/ADDING_TRANSLATIONS.en.md
at master · kleo-53/trikRuntime · GitHub. — URL: https://github.com/kleo-53/trikRuntime/blob/master/translations/ADDING_TRANSLATIONS.en.md (online; accessed: 2024-01-07).
- [24] Что такое локализация продуктов / Skillbox Media. — URL: <https://skillbox.ru/media/design/trudnosti-perevoda/> (online; accessed: 2023-12-13).