

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 23.Б15-мм

Поддержка системных и пользовательских тем оформления TRIK Studio

Кудря Александр Николаевич

Отчёт по учебной практике
в форме «Решение»

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

Санкт-Петербург
2025

Оглавление

Введение	3
1. Обзор	4
1.1. Общий механизм работы с темами в Qt	4
1.2. Текущая реализация тем в TRIK Studio	4
1.3. Недостатки текущей реализации	5
2. Постановка задачи	6
3. Реализация	7
3.1. Устранение зависимости от жёстко заданных значений .	7
3.2. Реализация поддержки корректного отображения SVG- иконок	8
3.3. Исправление проблем с отображением тёмной темы на Windows	9
Заключение	10
Список литературы	12

Введение

Более 45% школьников в России имеют нарушения зрения [3], такие как миопия, цветовая слепота или светобоязнь. Особенно важно не забывать об этом в цифровом обучении, где дети проводят много времени за экранами. Постоянное взаимодействие с экраном может усугублять зрительные проблемы и вызывать дискомфорт. Одним из решений этой проблемы является поддержка различных тем.

Тема приложения — это параметры визуализации, задающие внешний вид интерфейса. Поддержка тем обеспечивает доступность и комфорт для школьников с особенностями зрения, предлагая возможность изменения размеров интерфейса, высокую контрастность, тёмную и светлую темы, а также адаптацию цветовых сочетаний.

В данной работе под системной темой понимается тема, задаваемая настройками операционной системы и применяемая ко всем приложениям для обеспечения единообразия. Пользовательская тема применяется только к конкретному приложению под нужды пользователя.

TRIK Studio [1] — образовательная среда программирования с открытым исходным кодом, предназначенная для обучения программированию и робототехнике.

Графический интерфейс приложения построен на фреймворке Qt, который поддерживает работу с темами, что позволяет гибко адаптироваться под разные цветовые схемы. Но для полноценной поддержки системной темы требуется отдельная настройка каждого компонента и всего проекта. В версии TRIK Studio 2024.2 системная тема отображалась некорректно¹: некоторые элементы не изменялись при смене темы, иконки приложения не меняли цвет.

Для обеспечения работы системной темы в TRIK Studio внесены изменения, которые улучшили отображение всех элементов интерфейса в соответствии с пользовательскими настройками.

¹<https://github.com/trikset/trik-studio/issues/1712>

1. Обзор

В данном разделе будут рассмотрены работа с темами в Qt, текущая реализация адаптации под тему в TRIK Studio, а также проблемы, обнаруженные при изучении исходного кода и приложения.

1.1. Общий механизм работы с темами в Qt

В основе темизации² в Qt лежит объект QPalette, который хранит системные цвета для элементов интерфейса: цвет текста, цвет фона, цвет выделения и прочие цвета. Если всё в приложении опирается на QPalette, виджеты³ автоматически меняют свои цвета в соответствии с новой темой.

Также для темизации может использоваться механизм Qt Style Sheets (QSS), который позволяет определять стилизацию интерфейса, аналогично CSS⁴ в веб-разработке. С помощью QSS можно задавать цвета, шрифты, отступы и другие параметры оформления для отдельных виджетов или всего приложения в целом.

Отдельное внимание следует уделить и работе с SVG⁵-иконками. Qt не предоставляет автоматической возможности адаптации SVG в зависимости от темы.

Следует отметить, что при работе под Windows тёмная тема не всегда корректно распознаётся самим Qt [2]. Автоматическая подстройка под системную тёмную тему в версии Qt5.15 не работает.

1.2. Текущая реализация тем в TRIK Studio

В TRIK Studio первоначально предпринята попытка внедрения темизации через QSS. Однако эта реализация ограничена только стартовым

²Темизация – калька с английского термина *theming* (см. <https://en.wiktionary.org/wiki/theming>).

³Виджет — это базовый элемент графического интерфейса в Qt (см. <https://doc.qt.io/qt-6/~qwidget.html>)

⁴CSS (Cascading Style Sheets) — язык стилей, используемый преимущественно в веб-разработке для описания внешнего вида элементов, таких как цвета, шрифты и расположение (см. <https://www.w3.org/TR/CSS2/>).

⁵SVG (Scalable Vector Graphics) — язык разметки на основе XML для описания масштабируемой двумерной векторной графики.

экраном и использует жестко заданные цвета, ориентированные на светлую тему.

Остальные компоненты приложения в основном полагаются на стандартные механизмы Qt и корректно отображаются в соответствии с системной темой за исключением некоторых элементов, для которых цвета заданы явно для светлой темы.

Реализация адаптации иконок отсутствует.

1.3. Недостатки текущей реализации

Несмотря на частичное использование QSS, реализация темизации в TRIK Studio отсутствует. Ниже приведён перечень основных проблем.

1. Некоторые компоненты стартового экрана используют QSS с жёстко прописанными цветами, что препятствует автоматической адаптации к тёмной теме.
2. В отдельных виджетах цвета задаются напрямую, без обращения к QPalette, что приводит к несогласованности внешнего вида.
3. Отсутствует поддержка системной темы под Windows.
4. Нет механизма темизации SVG-иконок, поэтому некоторые иконки остаются некорректно окрашенными.

2. Постановка задачи

Цель работы — обеспечить адаптацию интерфейса TRIK Studio под системную тему, устранив применение фиксированных цветов и стилей. Для достижения данной цели были поставлены следующие задачи.

1. Проанализировать текущую реализацию стилизации в TRIK Studio, выделив места, требующие рефакторинга.
2. Устранить зависимость интерфейса от жёстко заданных значений в QSS и коде, заменив их на значения, получаемые через QPalette.
3. Реализовать поддержку корректного отображения SVG-иконок в соответствии с системной палитрой, обеспечив их адаптацию к светлой и тёмной темам.
4. Исправить проблемы с отображением тёмной темы на Windows.

3. Реализация

В данном разделе рассмотрено, как были решены поставленные задачи.

3.1. Устранение зависимости от жёстко заданных значений

Одной из ключевых проблем, выявленных в ходе изучения кода, было то, что в различных модулях TRIK Studio использовались фиксированные значения для цветов и размеров.

Например, в модулях, отвечающих за отрисовку графических элементов (таких как `edgeElement.cpp` и `editorViewScene.cpp`), вместо динамического получения цвета через `QPalette` применялись константы вроде `Qt::black`. Аналогичная ситуация наблюдалась и в QSS-файлах: в стилях стартового экрана явно прописывались цвета (например, `background-color: white;` и `color: black;`), что делало интерфейс негибким и неадаптивным к изменению системных настроек.

Для устранения этой зависимости было решено заменить фиксированные значения на динамические, получаемые из системной палитры:

- Все вызовы, где цвета задавались напрямую (например, в файлах `edgeElement.cpp`), константы заменены на значения, получаемые через `QApplication::palette().color`. Например, на рис.1 показано, как в `EdgeElement` фиксированный чёрный заменяется на системный цвет.
- В QSS-файлах (например, в `startTab/button.css` и `startTab/header.css`) вместо жёстко заданных значений внедрены выражения, например `@@Text@@` и `@@Window@@`, которые динамически заменяются функцией `loadSystemPallete()`. Эта функция считывает текущие цвета из `QApplication::palette()`, что позволяет стилям автоматически подстраиваться под системную тему.

```

EdgeElement::EdgeElement(const EdgeElementType &type,
    const Id &id, const models::Models &models)
    : Element(type, id, models)
    , mPenStyle(Qt::SolidLine)
    , mPenWidth(1)
    - , mPenColor(Qt::black) // Жёстко заданный цвет
    + , mPenColor(QApplication::palette().color(QPalette::Text))
    , mType(type)
    , mSrc(nullptr)
    , mDst(nullptr)

```

Рис. 1: Пример замены жесткого цвета на системный.

3.2. Реализация поддержки корректного отображения SVG-иконок

Изначально для создания иконок в TRIK Studio использовался прямой вызов `QIcon` с указанием пути к ресурсам (например, `"/icons/2dcomment.svg"`). Такой подход не обеспечивал автоматическую адаптацию цвета иконок к системной палитре, поэтому они всегда отображались в исходном виде.

Для решения этой проблемы были разработаны две функции в классе `AbstractItem`:

1. `loadThemedIcon`. Функция загружает SVG-иконку по указанному пути, создает `QPixmap` и с помощью `QPainter`, заполняет изображение заданным цветом. Это позволяет получить `QIcon`, цвет которого определяется параметром `color`.
2. `loadTextColorIcon` Это обёртка над первой функцией, которая автоматически получает цвет текста из палитры и применяет его для окраски иконки.

Данный подход обеспечивает однородное отображение SVG-иконок, их адаптацию к светлой и тёмной темам, а также улучшает визуальную гармонию интерфейса.

3.3. Исправление проблем с отображением тёмной темы на Windows

На платформе Windows TRIK Studio изначально не реагировало на активированную тёмную тему, поскольку Qt5.15, используемая приложением, не умеет понимать, когда в Windows Включена тёмная тема автоматически. Поэтому были добавлены две функции:

1. `windowsDarkThemeAvailiable`. Функция проверяет, поддерживает ли текущая версия Windows тёмную тему. В частности, если Windows 10 имеет обновление с номером микроверсии не ниже 17763, считается, что поддержка тёмной темы доступна.
2. `windowsIsInDarkTheme`. Эта функция считывает из реестра Windows значение ключа, отвечающего за темную тему.

Теперь если приложение запущено на Windows и активирована тёмная тема, создаётся новая `QPalette` с тёмными значениями для таких ролей, как `Window`, `WindowText`, `Base`, `Button` и прочих. Палитра создаётся на основе файла с настройками темы. Для этого была реализована функциональность загрузки палитры из файла.

Это решение обеспечивает применение тёмной палитры при запуске приложения, что позволяет улучшить визуальную согласованность интерфейса на Windows, даже если система не переключается динамически.

Заключение

В результате проделанной работы по улучшению поддержки тем в TRIK Studio достигнуты следующие результаты.

- Выявлены места, где использовались фиксированные значения цветов и размеров, что препятствовало адаптивному отображению интерфейса.
- Заменены жёстко заданные значения в коде и QSS на динамические, получаемые через QPalette. В результате, вместо использования фиксированных цветов, теперь применяется значение, соответствующее текущей системной палитре, что обеспечивает согласованное отображение элементов интерфейса.
- Реализована поддержка корректного отображения SVG-иконок. С помощью функций `loadThemedIcon()` и `loadTextColorIcon()` SVG-иконки автоматически адаптируются под цветовую схему, что позволяет им корректно отображаться как в светлой, так и в тёмной темах.
- Исправлены проблемы с отображением тёмной темы на платформе Windows. При запуске приложения производится проверка системных настроек Windows, и, если активирована тёмная тема, формируется соответствующая QPalette, обеспечивая применение тёмной палитры ко всему интерфейсу.

Для дальнейшего развития поддержки тем в TRIK Studio предлагается: создание встроенного инструмента для настройки пользовательских тем (цветов, шрифтов, стилей) с сохранением в файл конфигурации, переход на QSS для всего интерфейса, как это реализовано для стартового экрана, разработка стандартных высококонтрастных тем с настройкой размеров элементов для соответствия стандартам доступности WCAG⁶,

⁶WCAG (Web Content Accessibility Guidelines) — международные рекомендации по обеспечению доступности веб-контента (см. <https://www.w3.org/TR/WCAG21/>).

а также реализация переключения тем (светлой, тёмной, пользовательской) без перезапуска приложения с использованием сигналов и слотов Qt⁷.

Ссылка на пул-реквест с реализацией:

<https://github.com/trikset/trik-studio/pull/1896>

⁷<https://doc.qt.io/qt-6/signalsandslots.html>

Список литературы

- [1] Dmitry Mordvinov Yurii Litvinov Timofey Bryksin. TRIK studio: Technical introduction. — URL: <https://ieeexplore.ieee.org/document/8071326/authors#authors> (дата обращения: 6 ноября 2024 г.).
- [2] Does Qt support the dark theme for Windows 10? — 2019. — URL: <https://forum.qt.io/topic/101391/windows-10-dark-theme/4>.
- [3] Prevalence and associated factors of myopia in children and adolescents in Russia: the Ural Children Eye Study / Mukharram M. Bikbov, Gyulli M. Kazakbaeva, Albina A Fakhretdinova et al. — 2024. — URL: <https://bjo.bmj.com/content/108/4/593>.