

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 — Программная инженерия

Отчет по практике

РАЗРАБОТКА ПРИЛОЖЕНИЯ
«REKEY»

Выполнил:

студента 1 курса группы 22107

Ермаков Владимир Александрович

_____ *подпись*

Петрозаводск — 2020

Содержание

Введение	3
1 Требования к приложению	4
2 Проектирование приложения	5
3 Реализация приложения	6
Заключение	8

Введение

ReKey - программа для кеинга. Кеинг - это процесс совмещения двух и более изображений или кадров в пределах одной композиции. Основная задача при кеинге — убрать цветной фон и получить так называемую маску. Именно благодаря маске, специалист по композитингу сможет наложить нужный фон вместо цветного фона, на котором осуществлялась съемка.



Рис. 1: Кеинг наглядно

Цель проекта:

Получить практические навыки в проектировании, разработке приложения, постановке задач и планировании для реализации проекта

Задачи проекта:

1. Формирование требований для приложения
2. Выбор платформы для реализации приложения
3. Выбор языка программирования для реализации
4. Проектирование дизайна приложения
5. Разработка архитектуры
6. Реализация приложения
7. Анализ выполненных требований

1 Требования к приложению

Пользовательские требования:

- Интуитивный и удобный интерфейс
- Возможность выбора фона для создания маски
- Наличие гибких настроек и быстрого отображения результата
- Выдача результата, сравнимого по качеству с профессиональными инструментами кеинга

Основные функции:

- Импорт файла
- Настройка фона
- Предпросмотр результата
- Экспорт файла

Архитектура приложения:

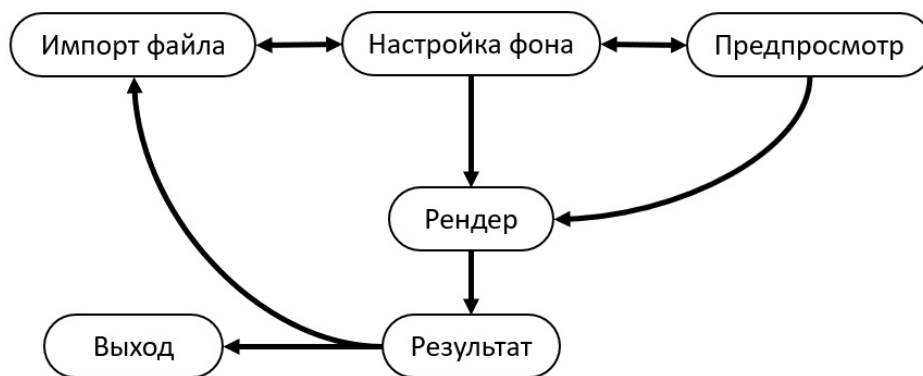


Рис. 2: Архитектура

2 Проектирование приложения

В приложении 2 основных модуля:

1. Модули реализации интерфейса:

- `main.qml` - корень интерфейса, связывает все модули
- `Toolbar.qml` - главное меню приложения
- `Editor.qml` - окно для выставления и редактирования настроек фона

2. Модули обработки изображения:

- `helper.cpp` - работа с видеофайлом
 - `load(path)` - импорт файла
 - `prerender(params)` - предпросмотр примененных настроек во выюпорте
 - `render(params)` - экспорт файла с применением настроек
- `convert.cpp` - конвертирование цвета отдельного пиксела
 - `convert(RGB)` - преобразование схемы RGB в HSV
 - `toRGB(String)` - преобразование строки ffffff в RGB
- `process_frame.cpp` - генерация маски отдельного кадра
 - `process(params)` - наложение фильтра на кадр

3 Реализация приложения

В качестве языков программирования были использованы QML (для написания интерфейса) и C++ (для реализации всей внутренней логики, связанной с вычислениями), использовалась библиотеки Qt и opencv

Библиотеки opencv:

- core
- imgproc
- imgcodecs
- videoio

Интерфейс приложения:



Рис. 3: Главное меню

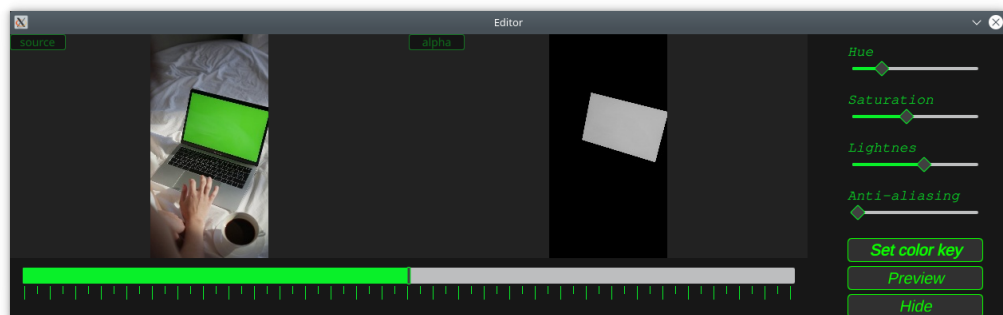


Рис. 4: Редактор

Для предпросмотра результата во вьюпорте используется подгрузка изображения, которое экспортируется из конкретного текущего кадра в изображение формата jpg

Итого:

- Файлов:

1. QML: 9

2. C++: 4

3. .JPG: 3

4. Другие: 5

- Функций: 12

- Время обработки одного кадра: 0.001 - 1.5 с.

Заключение

В ходе разработки были реализованы основные требования к приложению: удобная быстрая утилита по созданию маски для удаления фона. В перспективе возможно добавление таких функций как:

- Возможность создавать проекты (в файле проекта хранится ссылка на видео и настройки фона)
- Пипетка для выбора цвета фона
- Оптимизация алгоритма рендеринга
- Возможность как создания маски так и замены самого фона (необходима интеграция цветокоррекции видео)
- Возможность обрезки видео
- Улучшение сглаживания краев маски
- Переосмысление логики предпросмотра результата во вьюпорте без использования изображений