Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра электронных вычеслительных машин
Дисциплина: Программирование на языках высоккого уровня

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему

ASCII art program

Студент Е.С. Мелюх
Руководитель Е.В.Богдан

МИНСК 2023

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ

(подпись)
2023 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту *Мелюху Евгению Сергеевичу*
1. Тема проекта *«ASCII-ART»*
2. Срок сдачи студентом законченного проекта *15 декабря 2023 г.*
3. Исходные данные к проекту: изображения
4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)
1. Введение.
2. Задание.
3. Обзор литературы.
3.1. Обзор методов и алгоритмов решения поставленной задачи.
4. Функциональное проектирование.
4.1. Структура входных и выходных данных.
4.2. Разработка диаграммы классов.
4.3. Описание классов.
5. Разработка программных модулей.
5.1. Разработка схем алгоритмов (два наиболее важных метода).
5.2. Разработка алгоритмов (описание алгоритмов по шагам для двух методов).
6. Результаты работы.
7. Заключение
8. Литература
9. Приложения
5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)
1. Диаграмма классов.
2. Схема алгоритма `resized_image(Image *image,int new_height,int new_width).`
3. Схема алгоритма `ascii ( Image * image, std::string file_name).`
6. Консультант по проекту (с обозначением разделов проекта) Е.В.Богдан

7. Дата выдачи задания *15 сентября 2023 г.* 8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

1. Выбор задания. Разработка содержания пояснительной записки. Перечень графического материала – 15 %;

разделы 2, 3 – 10 %;

разделы 4 к –20 %;
разделы 5 к – 35 %;
раздел 6,7,8 – 5 %;
раздел 9 к – 5%;
оформление пояснительной записки и графического материала к 15.12.23 – 10 %
Защита курсового проекта с 21.12 по 28.12.23г.

РУКОВОДИТЕЛЬ Е.В.Богдан

_____

(подпись)

Задание принял к исполнению _____ Е.С.Мелюх

(дата и подпись студента)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современные возможности по разработке прикладного программного обеспечения с использованием языка высокого уровня C++ включают в себя следующие аспекты:

Объектно-ориентированное программирование (ООП), которое позволяет создавать программы, основанные на объектах, а не на процедурах. Это позволяет создавать более гибкие и масштабируемые Template Library), которые предоставляют широкий набор готовых компонентов для решения различных задач.

Использование современных инструментов разработки, таких как среды разработки (IDE), отладчики и компиляторы, которые позволяют ускорить процесс разработки и улучшить качество кода.

Применение современных методов разработки, таких как Agile и DevOps, которые позволяют создавать приложения быстрее и более эффективно.

Использование современных практик программирования, таких как TDD (Test-Driven Development) и CI/CD (Continuous Integration/Continuous Deployment), которые позволяют создавать более надежное и качественное изображение, состоящее из ASCII-символов, таких как буквы, цифры и специальные символы. Этот процесс позволяет создать уникальные и художественные представления изображений с помощью символов текста. Вот некоторые достоинства и преимущества преобразования изображения в ASCII-арт:

Творческое искусство: Преобразование изображения в ASCII-арт открывает новые возможности для художников и творческих личностей. Это позволяет им создавать уникальные и оригинальные произведения искусства с использованием символов текста.

Эффективное сжатие изображений: ASCII-арт может быть более компактным и занимать меньше места в сравнении с цветными изображениями, что полезно при передаче изображений через сеть или сохранении на диске.

Ретро-стиль и ностальгия: ASCII-арт имеет свои корни в ранних компьютерных искусствах и текстовых видеоиграх. Это может вызвать ностальгию у тех, кто вырос в эпоху, когда ASCII-арт был популярным.

Легкость создания: Процесс создания ASCII-арта может быть увлекательным и доступным для широкой аудитории. Для этого не требуется специализированного программного обеспечения, и множество онлайн-ресурсов и инструментов позволяют быстро создавать ASCII-арт.

Обучение программированию и креативности: Преобразование изображения в ASCII-арт может быть отличным способом введения новичков в мир программирования и компьютерного искусства. Это позволяет экспериментировать с различными алгоритмами и методами преобразования.

Что может заинтересовать людей в процессе преобразования изображения в ASCII-арт:

Художники и дизайнеры: Создание уникальных искусственных работ с использованием ASCII-арта может быть увлекательным и творческим процессом.

Любители ретро-компьютеров: ASCII-арт олицетворяет дух ранних компьютерных игр и искусства, что может привлечь поклонников старых игровых платформ и компьютеров.

Образовательные цели: Создание ASCII-арта может быть отличным способом обучения основам программирования и компьютерного искусства.

Процесс преобразования изображения в ASCII-арт основан на анализе цветов и яркости пикселей изображения и их соответствии символам ASCII в зависимости от их интенсивности. Существует множество алгоритмов, которые могут использоваться для этой цели, и они могут быть адаптированы для различных уровней детализации и творчества программное обеспечение.

Преобразование изображения в ASCII-арт (ASCII art) - это процесс, при котором цветное изображение или фотография преобразуется в символьное изображение, состоящее из ASCII-символов, таких как буквы, цифры и специальные символы. Этот процесс позволяет создать уникальные и художественные представления изображений с помощью символов текста. Вот некоторые достоинства и преимущества преобразования изображения в ASCII-арт:

Творческое искусство: Преобразование изображения в ASCII-арт открывает новые возможности для художников и творческих личностей. Это позволяет им создавать уникальные и оригинальные произведения искусства с использованием символов текста.

Эффективное сжатие изображений: ASCII-арт может быть более компактным и занимать меньше места в сравнении с цветными изображениями, что полезно при передаче изображений через сеть или сохранении на диске.

Ретро-стиль и ностальгия: ASCII-арт имеет свои корни в ранних компьютерных искусствах и текстовых видеоиграх. Это может вызвать ностальгию у тех, кто вырос в эпоху, когда ASCII-арт был популярным.

Легкость создания: Процесс создания ASCII-арта может быть увлекательным и доступным для широкой аудитории. Для этого не требуется специализированного программного обеспечения, и множество онлайн-ресурсов и инструментов позволяют быстро создавать ASCII-арт.

Обучение программированию и креативности: Преобразование изображения в ASCII-арт может быть отличным способом введения новичков в мир программирования и компьютерного искусства. Это позволяет экспериментировать с различными алгоритмами и методами преобразования.

Процесс преобразования изображения в ASCII-арт основан на анализе цветов и яркости пикселей изображения и их соответствии символам ASCII в зависимости от их интенсивности. Существует множество алгоритмов, которые могут использоваться для этой цели, и они могут быть адаптированы для различных уровней детализации и творчества.

1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Овладение практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта на языке высокого уровня C++ является ключевой целью для развития навыков программирования. Для достижения этой цели предлагается разработать приложение "ASCII Art Program" с использованием среды разработки Qt. Вот несколько общих целей, которые могут быть установлены для этого процесса обучения:

Загрузка изображения:

- Реализовать функционал загрузки изображения из файлов различных форматов, таких как JPEG, PNG, GIF и других популярных форматов.

Преобразование изображения в ASCII:

- Разработать алгоритм преобразования каждого пикселя изображения в соответствующий символ ASCII в зависимости от его яркости.
- Разработать алгоритм уменьшения размера изображения для удобного просмотра его.
- Обеспечить возможность выбора различных символов для представления разных уровней яркости.

Сохранение преобразованного изображения:

- Предоставить функционал сохранения полученной ASCII-графики в текстовые файлы.

Поддержка нескольких форматов изображения:

- Обеспечить возможность работы с изображениями различных форматов, учитывая особенности представления цвета и яркости в каждом из них.
- Реализовать соответствующие модули для обработки различных форматов изображений.

Интерфейс пользователя:

- Разработать удобный интерфейс для взаимодействия пользователя с программой, включая возможность выбора изображения.

2 ОБЗОР ЛИТЕРАТУРЫ

Перед началом разработки нужно было ознакомится с алгоритмом «ASCII-ART». При ознакомлении алгоритма был использован источник (сайт) [7].

ASCII-арт – это удивительное искусство, где изображения создаются с использованием символов ASCII (American Standard Code for Information Interchange). Вот шаги алгоритма:

Выбор Изображения: Всё начинается с выбора изображения. Это может быть фотография, рисунок или любой другой графический материал.

Преобразование в Оттенки Серого: Чтобы упростить процесс, изображение преобразуется в оттенки серого. Это помогает снизить количество цветов и легче перевести их в символы ASCII.

Разделение на Блоки: Изображение делится на блоки пикселей, каждый из которых заменяется символом ASCII в зависимости от яркости блока;

Выбор Символов ASCII: Определяю набор символов ASCII, которые будут использованы для замены блоков. Использую различные символы, такие как '@', '#', '8', '%', и конечно, пробел для представления разных оттенков.

Создание ASCII-арта: Каждый блок пикселей заменяется соответствующим символом ASCII, и появляется уникальное ASCII-арт изображение.

Отображение или Сохранение: Готовое ASCII-арт изображение может быть отображено на экране или сохранено в файле для дальнейшего использования.

Простота: Создание ASCII-арта – процесс, доступный каждому. Не требуется специальных навыков или сложных инструментов.

Легкость Восприятия: ASCII-арт легко читается в текстовой форме, что делает его удобным для создания креативных текстовых изображений;

Легкость Передачи: ASCII-арт может быть вставлен в текстовые сообщения, письма или код, что делает его легко передаваемым.

Так же для разработки алгоритма уменьшения размера изображения был использован сайт [11] - англоязычный ресурс, где математически описывается алгоритм по уменьшению размера изображения.

Так же при разработке была использована официальная документация Qt project[8]. Qt — это мощный фреймворк для разработки кроссплатформенных приложений на C++. Он предоставляет широкий набор инструментов и библиотек для создания графических интерфейсов, обработки событий, работы с сетью, базами данных и другими задачами. Вот краткое описание документации Qt:

Официальная документация: Сайт Qt: Официальный сайт Qt содержит обширную документацию, включая руководства, API-справочники, примеры кода и другие материалы.

Руководства и Введение в Qt:

Getting Started: Раздел "Getting Started" в документации обычно предоставляет информацию о том, как установить Qt, настроить среду разработки и создать простое приложение. Overview: Введение в фреймворк, его основные концепции и принципы.

Создание графического интерфейса: Qt Widgets: Информация о виджетах Qt, базовых элементах управления, таких как кнопки, поля ввода и другие. Qt Quick и QML: Документация о создании интерфейсов с использованием декларативного языка QML и фреймворка Qt Quick.

Работа с сетью и базами данных: Qt Network: Инструменты для работы с сетью, включая HTTP-запросы, сокеты и другие. Qt SQL: Информация о работе с базами данных, включая поддержку различных СУБД.

Многозадачность и Параллелизм: Qt Concurrency: Раздел, посвященный поддержке многозадачности и параллелизма в Qt.

Межплатформенная разработка: Platform Notes: Рекомендации и особенности для кроссплатформенной разработки на разных операционных системах. Deployment: Инструкции по развертыванию Qt-приложений на различных платформах.

Примеры кода и Учебные проекты:Qt Examples: Обширный набор примеров кода для различных компонентов Qt.Qt Tutorials: Учебные проекты и туториалы, позволяющие освоить различные аспекты фреймворка.

Форумы и Сообщества: Qt Forum: Онлайн-форумы, где разработчики могут задавать вопросы, делиться опытом и получать поддержку от сообщества Qt.

Так же существуют и аналоги `ascii-art`:

- ANSI Art: Это форма искусства, подобная ASCII-арту, но использующая символы и цвета из таблицы ANSI. ANSI-арт часто встречается в текстовых файлах или на BBS (Bulletin Board Systems) из прошлого;
- Unicode Art: Использует символы из таблицы Unicode для создания иллюстраций. Это расширяет возможности создания изображений с использованием более широкого набора символов;
- ASCII-символы в комбинации с текстом: Некоторые художники используют ASCII-символы в сочетании с текстовыми элементами, чтобы создавать уникальные и креативные изображения;
- Text-Based Emoticons (Kaomoji): Это форма искусства, включающая создание изображений с использованием текстовых символов для передачи эмоций. Примеры включают смайлики и "каомодзи";
- Pixel Art: Хотя это не совсем текстовое искусство, пиксельное искусство использует блоки пикселей для создания изображений, часто в стиле, напоминающем ASCII-арт;
- ANSI Escape Code Art: Использует escape-коды ANSI для управления цветами и форматированием в текстовом выводе, что может быть использовано для создания цветных и стилизованных текстовых изображений;

Так же существует множество уже созданных приложений одними из низ являются:

- ASCII Art Studio: Это приложение для Windows, которое предоставляет инструменты для создания ASCII-артов и ANSI-артов. Оно поддерживает различные форматы файла и имеет редактор с расширенными функциями.
- ASCII Art Generator (Android): Это мобильное приложение для платформы Android, которое позволяет пользователям создавать ASCII-арт на своих мобильных устройствах.
- Textaizer: Это приложение для создания ASCII-артов, которое позволяет преобразовывать изображения в текстовый вид. Оно поддерживает различные стили и настройки.
- Jave (Just Another Variation of Emacs): Это текстовый редактор, который поддерживает рисование графики с использованием ASCII-символов. Он предоставляет режим "artist mode" для создания и редактирования ASCII-артов.

2.1 Обзор методов и алгоритмов решения поставленной задачи

Для решения задачи был выбран язык программирования C++ и методология объектно-ориентированного программирования.

В процессе разработки программы были использованы различные возможности языка C++, которые будут описаны ниже:

Для хранения значения пикселей был выбран вектор. Для храненнния значения цветов пикселй была написана структура `Pixel`, в которой были использованы поля для хранения красного, зеленого и синего цветов соответсвенно и перегружены операторы для удобного ображения к полям и использования значений для расчета интерполяции.

Для описания изображения был написан абстарктный класс `Image` и его наследники `PPMImage, BMPImage, OtherImage.`

Для считывания данных о изображениях были написаны методы для считывания.

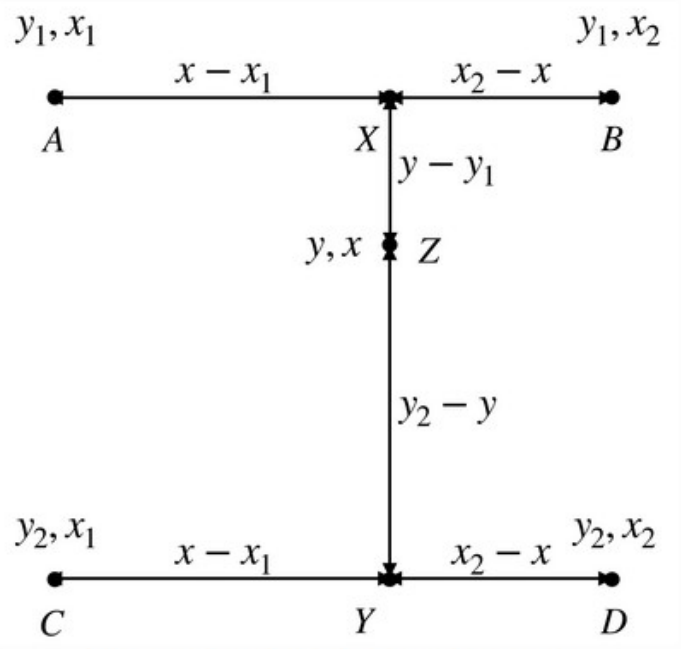
Для считывания формата `bmp` были написаны структуры для поддержки формата изображения `BMPImageHeader` и `BMPFileHeader`, в которых были описаны все поля для хранения файла.

Для преобразования в черно-белый, изменения размера изображения и перевода в `ascii-art` был реализован класс `Convert.`

Так же были реализованы классы для отображения графического интерфейса и более удобного взаимодействия с ним.

Алгоритм уменьшения размера изображения включает в себя получения новых размеров и получение доступа ко всем данным изображения.Создается массив нового размера, в который будут записаны интерполляционные пиксели, это достигается за счет вычисления координат соответствующего пикселя в исходном изображении, на основе координат пикселя в измененном изображении, и последующего выполнения интерполяции между ближайшими пикселями в исходном изображении.

На картинке 2.1.1 представлен визуализированный анализ соседей по осям X и Y.



Bilinear interpolation between four points.

2.1.1- анализ соседей

На изображения 2.1.2, 2.1.3, 2.1.4 представлены формулы расчета интерполяции.

$$X = A(1 - w_x) + Bw_x$$
$$Y = C(1 - w_x) + Dw_x$$

2.1.2- формула расчета интерполяции по оси X и Y.

$$\begin{aligned} Z &= X(1 - w_y) + Yw_y \\ &= A(1 - w_x)(1 - w_y) + Bw_x(1 - y_y) + C(1 - w_x)w_y + Dw_xw_y \end{aligned}$$

2.1.3 — расчет интерполяции между двумя интерполированными значениями.

$$w_x = \frac{x-x_1}{x_2-x_1} \text{ and } w_y = \frac{y-y_1}{y_2-y_1}.$$

2.1.4 — расчет координатных весов.

Перевод изображения в `ascii` основан на вычисления яркости пикселя и заменой его соответсвующим симоволом, из массива символов.Для определения индекса в массиве символом используется метод для вычисления яркости пискеля. На рисунке 2.1.5 представлена формула для расчета индекса.

$$\text{map}(value, start1, stop1, start2, stop2) = \left(\frac{value-start1}{stop1-start1}\right) \times (stop2 - start2) + start2$$

2.1.5 — формула расчета значения пикселя.

3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В данном разделе описываются входные и выходные данные программы, диаграмма классов, а также приводится описание используемых классов и их методов.

3.1 Структура входных и выходных данных

В качестве входных данных получаем изображение и получаем ascii-art.

3.2 Разработка диаграммы классов

Диаграмма классов данной работы показана в приложении А.

3.3 Описание классов

3.3.1 Класс Image

`Image` – это абстрактный класс для описания базовой структуры хранения изображения и работы с ним.

Описание полей класса:

`int m_width`- приватное поле для хранения ширины изображения.

`int m_height`-приватное поле для хранения высоты изображения.

`std::vector<std::vector<Pixel>> m_pixels`- приватный вектор для хранения пикселей изображения.

Описание методов:

`virtual std::vector<std::vector<Pixel>> get_pixels() const =0` -метод для получения вектора пикселей.

`virtual void set_pixels(const std::vector<std::vector<Pixel>> &pixels) =0 метод для установления нового значения вектора пикселей.`

`virtual int get_width() const =0` метод для получения ширины изображения.

`virtual int get_height() const =0` метод для получения высоты изображения.

`virtual void set_width(const int &new_width) =0` установить нвоое значение ширины изображения.

`virtual void set_height(const int &new_height) =0` установить новое значение ширины изображения.

3.3.2 Структура BMPFileHeader

`BMPFileHeader-` структура для описания заголовка файла.

Описание полей структуры:

`char m_magic[2]` массив символов для хранения сигнатуры изображения.

`unsigned int m_size` переменная для хранения размера всего изображения.

`unsigned int m_reserved` переменная для хранения зарезервированного места в файле .

`unsigned int m_offset` переменная для хранения смещения.

3.3.3 Структура BMPImageHeader

`BMPImageHeader` – структура для описания заголовка изображения.

Описание полей структуры:

`unsigned int m_header_size` -перменная для хранения размера заголовка.

`int m_width` — переменная для хранения ширины изображения.

`int m_height` - переменная для хранения высоты изображения.

`unsigned short m_planes` - переменная для хранения количество плоскостей изображения.

`unsigned short m_bpp` - переменная для хранения количества пикселей на метр в изображении.

`unsigned int compression` - переменная для хранения сжатия изображения.

`unsigned int m_image_size` - переменная для хранения размера самих пикселей .

`int m_x_ppm` - переменная для хранения пикселей на метр по горизонтали.

`int m_y_ppm` - переменная для хранения пекслей на метр по вертикали .

`unsigned int m_colors` - переменная для хранения количества цветов.

`unsigned int m_important_colors`- переменная для хранения важных цветов.

3.3.4 Класс BMPImage

`BMPImage` – класс описывающий `bmp` файл.Наследуется от класса `Image`.

Описание методов:

`std::vector<std::vector<Pixel>> get_pixels() const` - метод для получения вектора пикселей.

`void set_pixels(const std::vector<std::vector<Pixel>> &pixels)` метод для установления нового значения вектора пикселей.

`int get_width() const` метод для получения ширины изображения.

`int get_height() const` метод для получения высоты изображения.

`void set_width(const int &new_width)` установить нвоое значение ширины изображения.

`void set_height(const int &new_height)` установить новое значение ширины изображения.

3.3.5 Класс PPMImage

`PPMImage` — класс описывающий `ppm` файл. Наследуется от класса `Image`.

Описание полей класса:

`int m_max_color` переменная для хранения максимального количества цветов.

`std::string m_magic-` для хранения сигнатуры файла.

`char m_last_line` – последняя строка в файле перед пикселями.

Описание методов:

`std::vector<std::vector<Pixel>> get_pixels() const` - метод для получения вектора пикселей.

`void set_pixels(const std::vector<std::vector<Pixel>> &pixels)` метод для установления нового значения вектора пикселей.

`int get_width() const` метод для получения ширины изображения.

`int get_height() const` метод для получения высоты изображения.

`void set_width(const int &new_width)` установить нвоое значение ширины изображения.

`void set_height(const int &new_height)` -установить новое значение ширины изображения.

`void read_image(const std::string & file_name)` – метод для считывания ppm формата.

3.3.6 Структура Pixel

`Pixel` – структура для описания пикселя изображения.

Описание полей структуры:

`unsigned char m_blue` — переменная для хранения значения синего цвета.

`unsigned char m_green` — переменная для хранения значения зеленого цвета.

`unsigned char m_red` — переменная для хранения значения зеленого цвета.

Описание методов:

`Pixel operator +(const Pixel &other) const` — перегруженный оператор для сложения.

`Pixel operator-(const Pixel & other) const` — перегруженный оператор для вычитания.

`Pixel operator*(float scalar) const` — пергруженный оператор для умножения элементов структуры на число.

`bool operator ==(const Pixel &other) const` — перегруженный оператор для проверки равенства.

`bool operator!=(const Pixek &other) const` — перегруженный оператор для указания не равно.

3.3.7 Класс BMPImageProcess

BMPImageProcess- класс для обработки bmp файла.

Описание полей класса:

BMPFileHeader m_bmpFileHeader – переменная,которая хранит данные заголовка BMPFileHeader.

BMPImageHeader m_bmpImageHeader – переменная, которая хранит данные заголовка BMPImageHeader.

std::vector<std::vector<Pixel>> m_pixels – вектор, который хранит значения пикселей.

Описание методов:

std::vector<std::vector<Pixel>> read_data_of_pixels(std::fstream &file) const – метод, который считывает данные о пикселях.

void read_image(const std::string &file_name) – метод для считывания данных о изображении.

std::vector<std::vector<Pixel>> get_pixels() const – метод для получения массива пикселей.

int get_width() const – метод для получения ширины изображения.

int get_height() const – метод для получения высоты изображения.

3.3.8 Класс Convert

Convert – класс для преобразования изображения в черно-белый, уменьшения его размера и перевода его в ascii-art.

Описание методов:

static void ascii(const Image* image,const std::string&file_path) – метод для перевода изображения в ascii-art и записи его в файл.

static void grayscale(Image *image) – метод для перевода изображения в черно-белый.

static std::vector<std::vector<Pixel>> resized_image(Image *image,int new_heigth,int new_width) – метод для изменения размера изображения.

static float map(float value,float start1,float stop1,float start2,float stop2) – меод для расчета индекса элемента ascii символа в массиве символов.

3.3.9 Класс SecondScreen

`SecondScreen` — класс для отображения второго экрана программы и работы с самой программой.Наследуется от Qwidget.

Описание полей класса:

`Ui::SecondScreen *ui` – указатель на объект `Ui`. Предназначен для работы с виджетами.

`QgraphicsScene *scene` – указатель на объект `QGraphicsScene`. Предназначен для работы со сценами.

Описание методов:

void remove_item(const Qstring &text) — удалить элемент из списка.

`void on_list_item_clicked(QListWidgetItem *item)` – выбрать элемент в списке.

`void on_pushButton_clicked()` - выбрать файл из проводника.

`void on_pushButton_2_clicked()` - перевести в ascii-art.

`bool is_image(const Qstring &file_path)` – проверить на формат изображения.

`void load_image(const Qstring &file_path)` – загрузить изображение.

3.3.10 Класс MainWindow

`MainWindow` – класс для отображения пригласительного экрана. Наследутеся от QMainWindow.

Описание полей класса:

`Ui::MainWindow *ui` -указатель на объект `Ui`. Предназначен для работы с виджетами.

`SecondScreen *secondScreen`- указатель на объект SecondScreen.

`QStackedWidget *stackWidget` – указатель на объект `QstackedWidget`.

Описание методов:

`void on_pushButton_clicked()` - метод для обработки нажатия и переключения на другой экран.

3.3.11 Класс CustomWindow

`CustomWindow` — класс для представления элемента QlistWidgetItem. Наследуется от QWidget.

Описание полей класса:

`Ui::CustomWindow *ui` -указатель на объект `Ui`. Предназначен для работы с виджетами.

Описание методов:

`void set_text(const Qstring &text)` -метод для установления нового текста.

`Qstring get_text()` - метод для получения текста.

`void send_remove_item(const Qtring &text)` – метод для удаления элемента.

`void close_button_clicked()` - метод для обработки нажатия кнопки удаления из списка элемента.

3.3.12 Класс OtherImage

`OtherImage` — класс описывающий иные форматы файлов. Наследуется от класса `Image.`

`std::vector<std::vector<Pixel>> get_pixels() const` - метод для получения вектора пикселей.

`void set_pixels(const std::vector<std::vector<Pixel>> &pixels)` метод для установления нового значения вектора пикселей.

`int get_width() const` метод для получения ширины изображения.

`int get_height() const` метод для получения высоты изображения.

`void set_width(const int &new_width)` установить нвоое значение ширины изображения.

`void set_height(const int &new_height)` -установить новое значение ширины изображения.

`std::vector<std::vector<Pixel>> read(const Qstring & file_path)` – метод для считывания других форматов .

4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

4.1 Разработка алгоритма `grayscale(Image *image)`

Метод `grayscale(Image *image)` класса `Convert` переводит изображение в черно-белый.

Шаг 1.Получение `2D-вектора` `pixels` из объекта `Image`.

Шаг 2.Создание нового `2D-вектора` `grayscale_pixels` с размерами изображения.

Шаг 3.Перебор каждого пикселя изображения с использованием двух вложенных циклов.

Шаг 4.Для каждого пикселя рассчет нового значения оттенка серого (`gray`) по формуле: `gray = 0.3 * red + 0.12 * blue + 0.58 * green`.

Шаг 5.Присвоение gray каждому цветовому каналу (красному, зеленому и синему) в `grayscale_pixels`.

Шаг 6.Установка новых пикселей в объект `Image`:

Шаг 7.Установка вектора `grayscale_pixels` в качестве новых пикселей объекта `Image`.

4.2 Разработка алгоритма `read_data_of_pixels (std::ifstream &file)`

Метод `read_data_of_pixels(std::ifstream &file) const` `метод класса` BMPImageProcess `для чтения данных о пикселях`.

Шаг 1.Вычисление размера строки row_size в байтах. Размер строки зависит от ширины изображения и числа байт на пиксель (m_bmpImageHeader.m_width * (m_bmpImageHeader.m_bpp / 8)).

Шаг 2. Вычисление отступа строки row_padding. Для выравнивания размера строки до 4 байт добавляется дополнительный байт, если размер строки не кратен 4.

Шаг 3.Вычисление смещения pixel_offset, откуда начинаются данные пикселей в файле. Смещение рассчитывается с учетом заголовка файла (m_bmpFileHeader.m_offset) и отступа строки.

Шаг 4.Перемещение указателя файла на позицию pixel_offset с использованием file.seekg.

Шаг 5.Создание 2D-вектора image_pixels для хранения пикселей изображения.

Шаг 6.Выделение памяти для буфера buffer, который будет использоваться для чтения данных из файла.

Шаг 7.Использование двух вложенных циклов для обхода пикселей изображения.

Шаг 8.Для каждого пикселя чтение данных пикселя в буфер из файла с использованием file.read.

Шаг 9.Присвоение значений компонент цвета пикселя из буфера в соответствующие элементы вектора image_pixels.

Шаг 10.Освобождение выделенной памяти для буфера с использованием delete[] buffer.

Шаг 11.Возврат вектора image_pixels, который теперь содержит данные о пикселях изображения.

5 РЕЗУЛЬТАТ РАБОТЫ

На рисунке 5.1 изображена начало работы программы. При старте программы встречает привественное окно с описанием идеи программы и кнопкой с переходом на следующий экран ,где происходит работа самой программы.

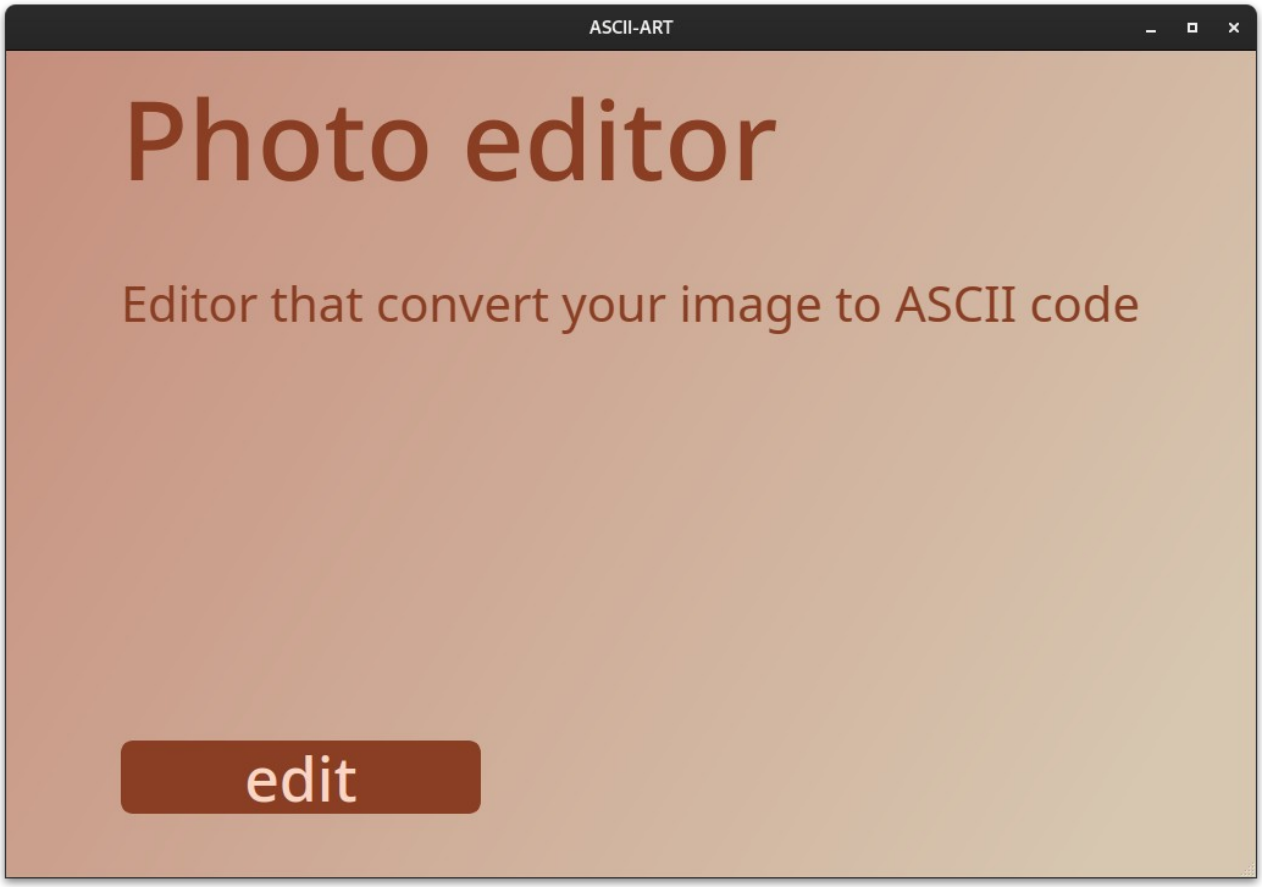


Рисунок 5.1 — Начало работы программы

На рисунке 5.2 показана работа кнопки перехода с первого экрана на второй и сам экран для работы с изображениями. В нем представлены кнопки для выбора файла из проводника и перевода в `ascii-art`. Также лист для представления списка загруженных файлов и графическое поле для просмотра картинкию.

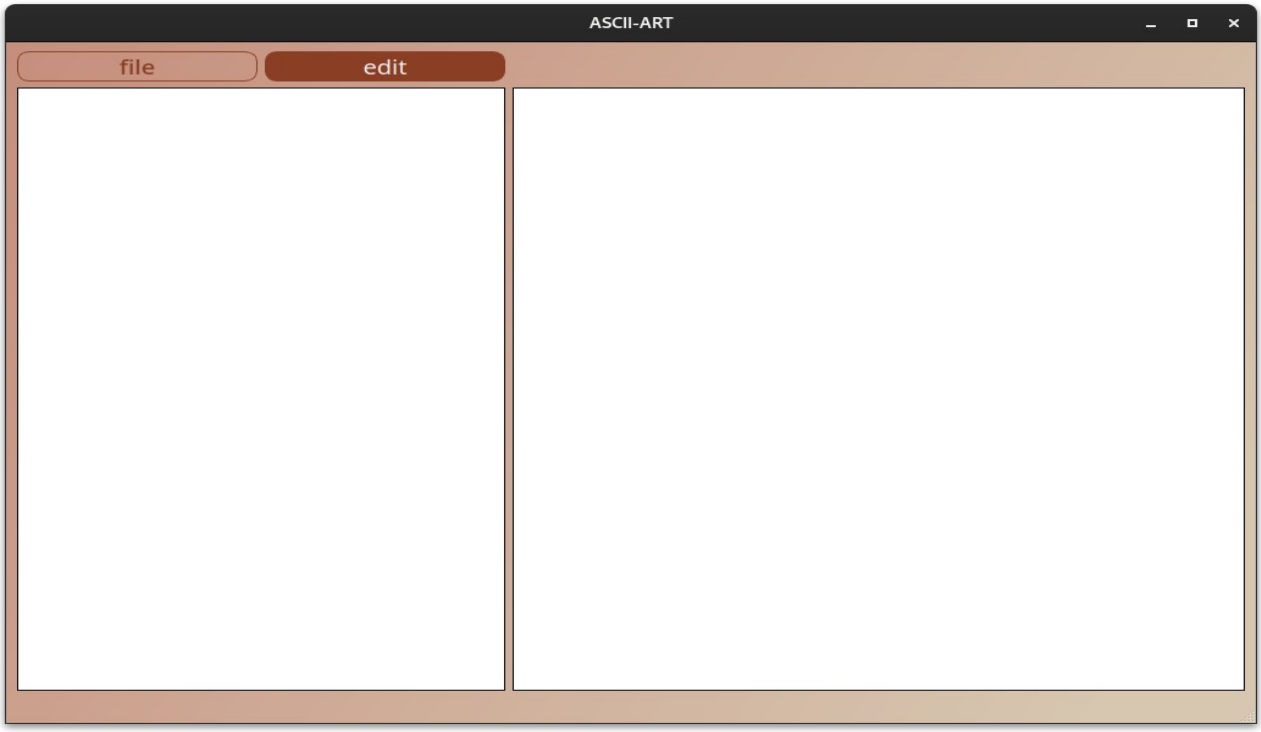


Рисунок 5.2 — Второй экран для работы с изображением

На рисунке 5.3 изображен открытый проводник, предоставляющий удобный доступ к файловой системе компьютера. Пользователь может свободно перемещаться по различным папкам и директориям, а также выбирать необходимые файлы для работы. Интерфейс проводника предоставляет удобные инструменты навигации и поиска, делая процесс выбора файла более эффективным и интуитивно понятным.

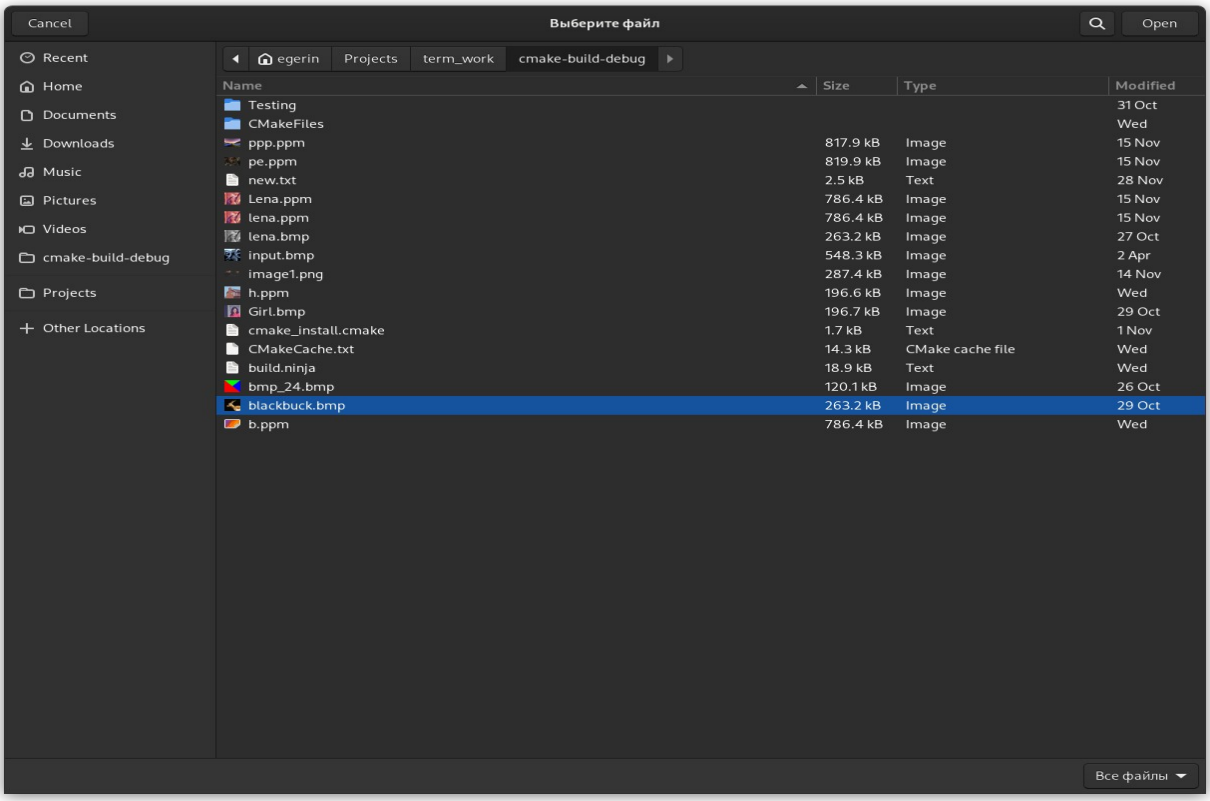


Рисунок 5.3 — Открытие проводника для выбора файла

На рисунке 5.4 представлен процесс добавления файла в лист и его последующее открытие в графическом поле. Пользователь, используя интерфейс, выбирает опцию добавления файла, после чего ему предоставляется доступ к файловой системе. После выбора необходимого файла и подтверждения, файл добавляется в лист для последующего редактирования. Далее, пользователь может открыть добавленный файл в графическом поле.

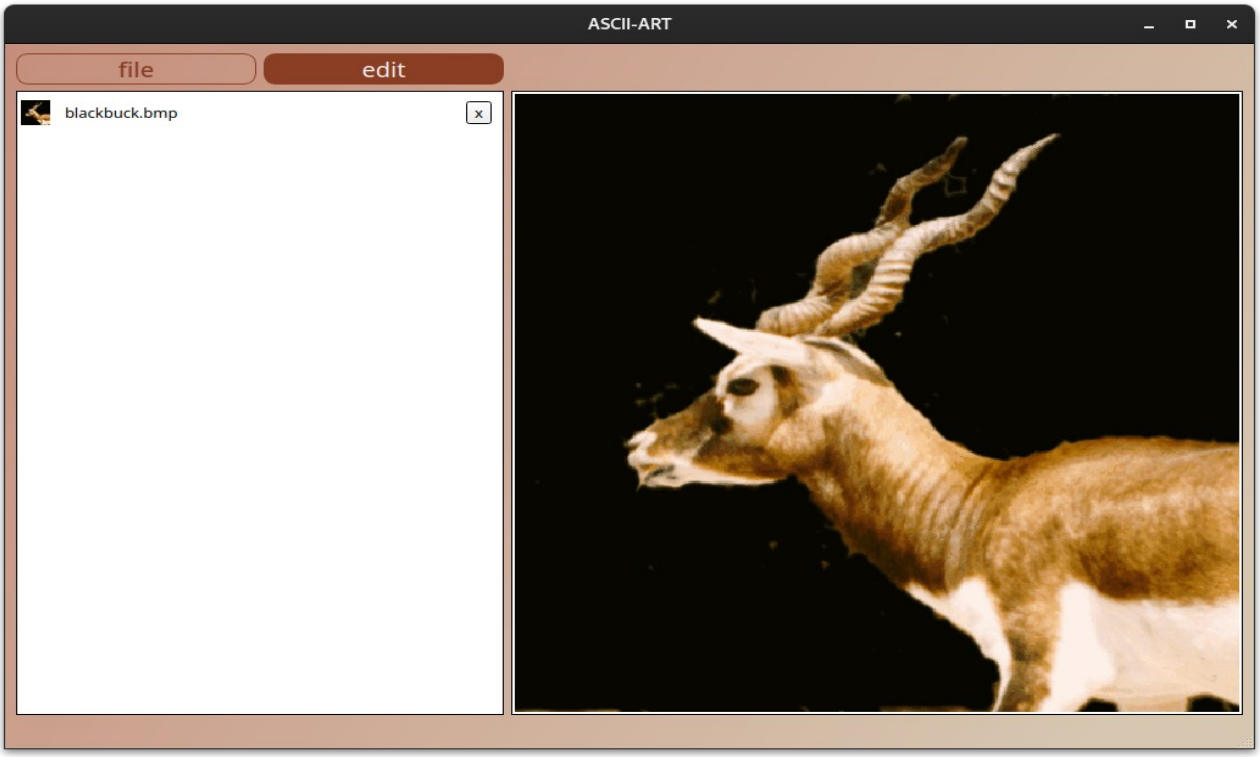


Рисунок 5.4 — Добавление файла в лист и открытие его в графическом поле.

На рисунке 5.5 изображено предупреждение о повторном открытии файла. Этот информационный диалоговый блок появляется, когда пользователь пытается открыть файл, который уже открыт в программе. Предупреждение предостерегает пользователя от возможных конфликтов при одновременном добавления одного и того же файла. Диалог предоставляет пользователю необходимые опции, такие как отмена операции, открытие файла в режиме "только для чтения" или продолжение открытия файла. Это обеспечивает гибкость в управлении файлами и предостерегает от потенциальных проблем, связанных с одновременным доступом к одному файлу со стороны разных пользователей

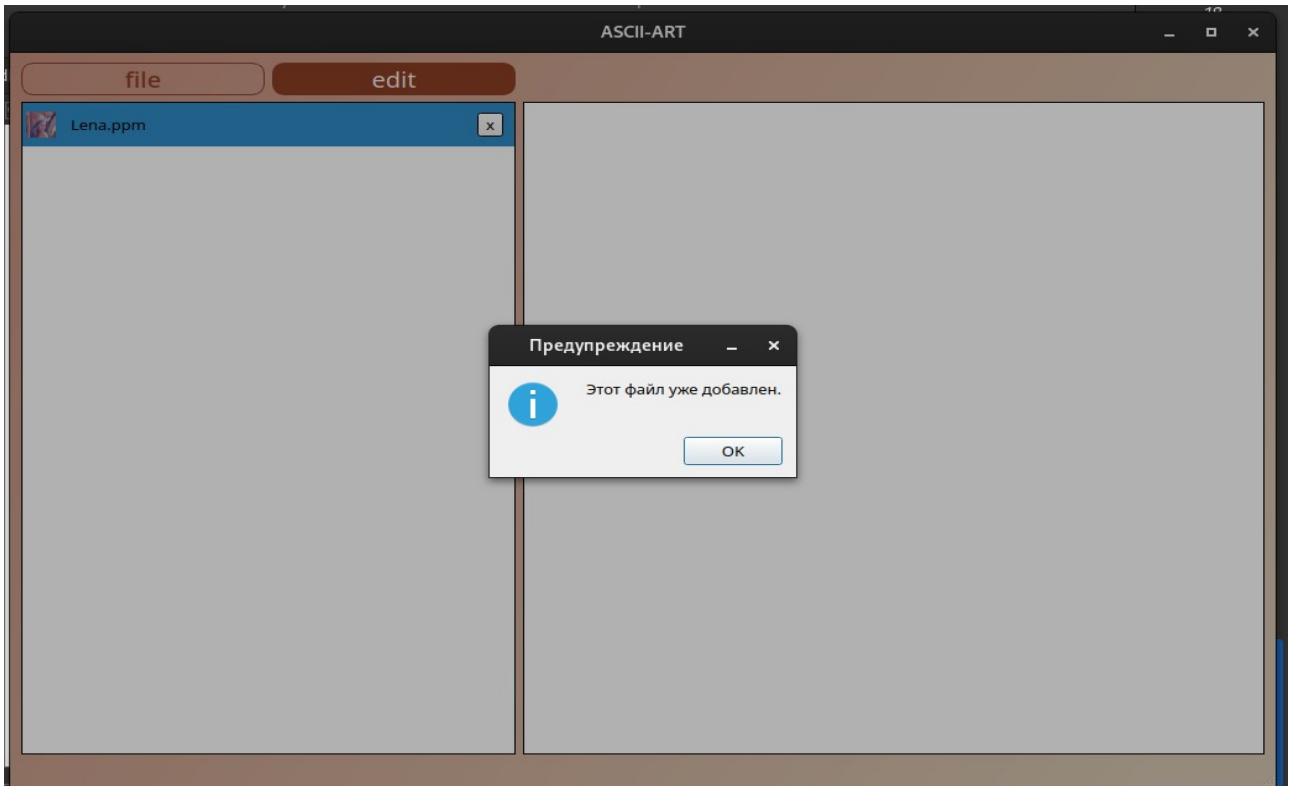


Рисунок 5.5 — Предупреждение о повторном откртии файла.

На рисунке 5.6 продемонстрирован процесс удаления файла из листа с использованием кнопки удаления. После подтверждения выбора, файл удаляется из листа. При удалении файла из листа, открытый в данный момент файл в графическом поле будет автоматически закрыт. Это предотвращает возможные конфликты и обеспечивает корректное управление ресурсами

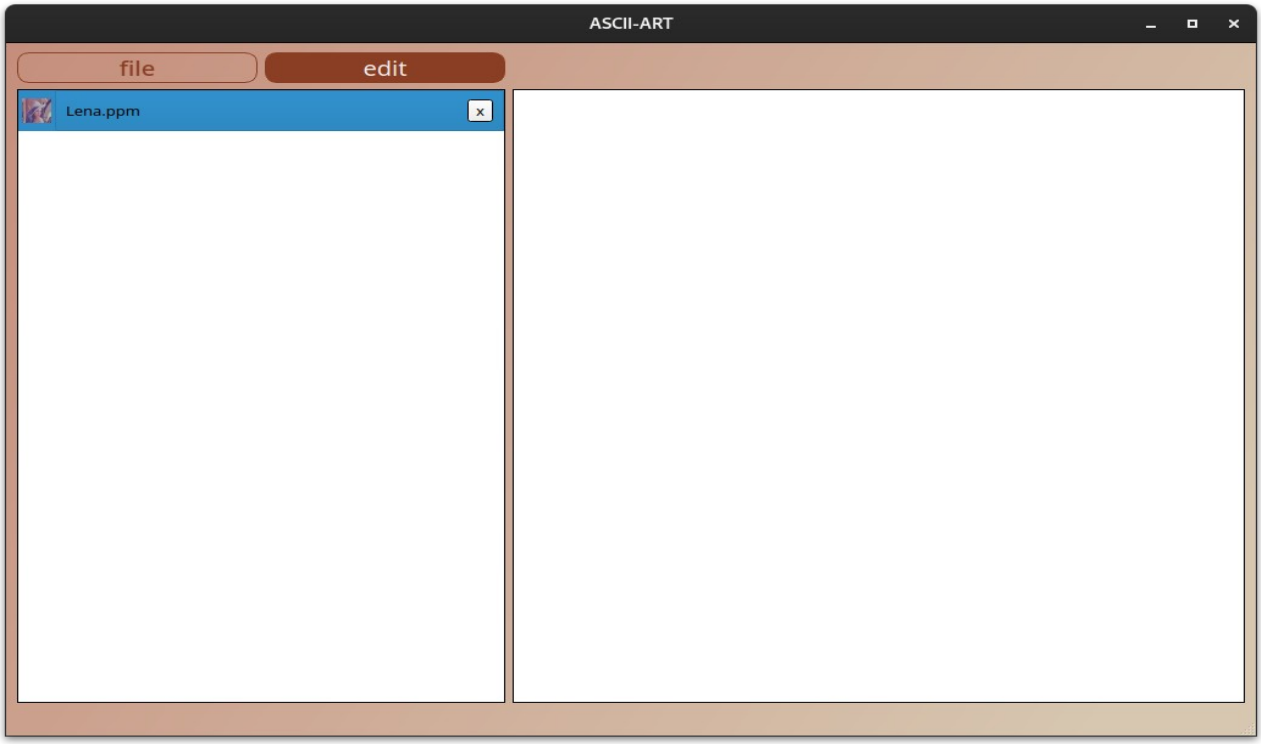


Рисунок 5.6 — Удаление файла из листа .

На рисунке 5.7 представлен процесс выбора файла и открытия его для редактирования. Пользователь взаимодействует с интерфейсом, выбирает необходимый файл, а затем нажимает кнопку `edit`. После этого файл, переведнный в `ascii-art`, открывается в соответствующем редакторе.

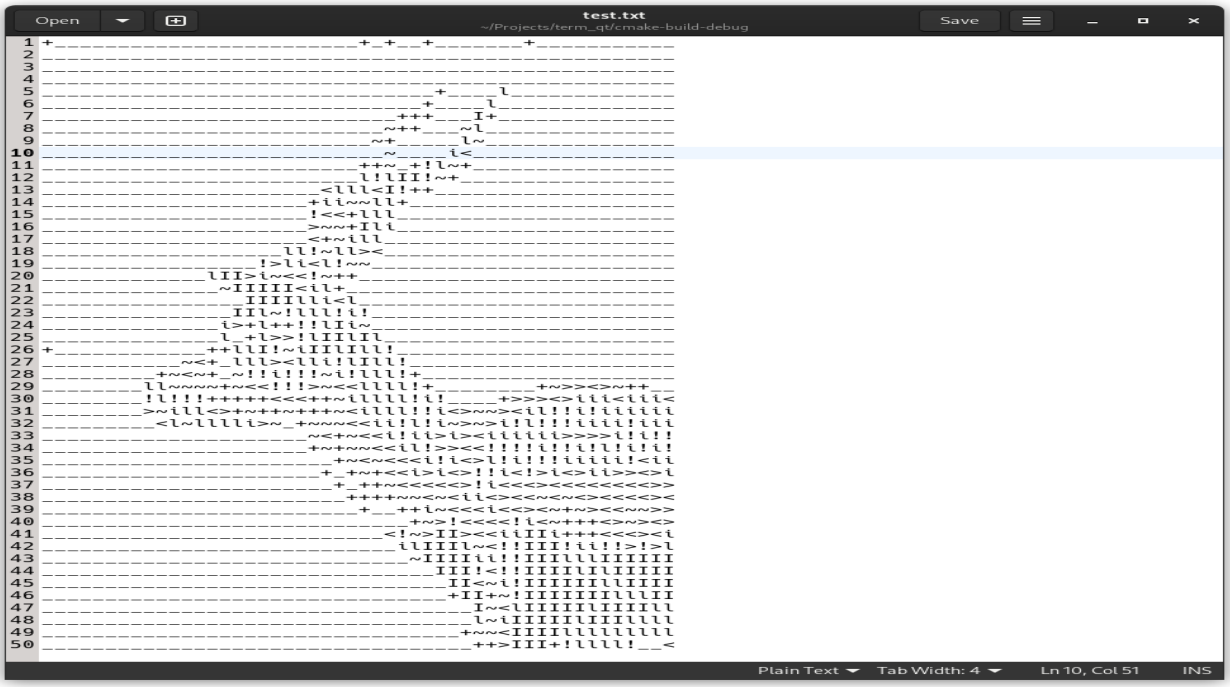


Рисунок 5.7 — Перевод в ASCII-ART .

Заключение

В результате выполнения курсовой работы, посвященной программам ASCII Art, были достигнуты значительные результаты и приобретены важные навыки в области обработки изображений и программирования на языке C++. Работа включала использование языка C++ в связке с фреймворком Qt, а также освоение навыков парсинга различных типов изображений без привлечения сторонних библиотек.

Использование C++ и Qt: Применение языка программирования C++ обеспечило высокую производительность и гибкость при разработке программы. Фреймворк Qt упростил создание графического интерфейса, обеспечивая кросс-платформенность и богатый инструментарий для визуализации данных. Это позволило создать удобный и интуитивно понятный пользовательский интерфейс, что является важным элементом успешной программы.

Навыки парсинга изображений: Освоение навыков парсинга различных форматов изображений без использования сторонних библиотек расширило функциональность программы. Реализованные алгоритмы позволяют эффективно обрабатывать изображения в форматах BMP и других, открывая новые перспективы для работы с разнообразными наборами данных.

Программирование ASCII Art: Основной задачей работы было создание ASCII Art изображений. Разработанные алгоритмы, включая методы билинейной интерполяции для изменения размеров изображения, позволяют эффективно создавать ASCII Art на основе входных изображений. Реализованный механизм преобразования пикселей в ASCII-символы, учитывая интенсивность цвета, добавляет интересные и креативные возможности для создания ASCII Art.

Итоги и перспективы: Эта курсовая работа стала важным этапом в понимании принципов работы с изображениями и применении алгоритмов ASCII Art в контексте программирования на C++. Полученные знания и опыт могут быть успешно применены в будущих проектах, таких как разработка графических редакторов, фильтров изображений и других приложений, связанных с обработкой изображений.

Вывод: В ходе курсовой работы были расширены навыки программирования на C++, изучены особенности работы с графикой и

создания графических интерфейсов с использованием фреймворка Qt. Полученные навыки и знания предоставляют отличную основу для дальнейших исследований и разработок в области компьютерного зрения и графики.

Список литераутры

[1] "ASCII Art: A Beginner's Guide" by Arnold Robbins

[2] The Art of Computer Programming" by Donald E. Knuth

[3] "Effective C++" by Scott Meyer

[4] "Object-Oriented Analysis and Design with Applications" by Grady Booch

[5] "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stei

[6] "UML Distilled: A Brief Guide to the Standard Object Modeling Language" by Martin Fowler

[7] Generate ASCII Art – A Simple. [Электронный ресурс].Режим доступа: https://www.devleader.ca/2023/08/25/generate-ascii-art-a-simple-how-to-in-c/. Дата доступа 4.12.2023.

[8] Сайт Qt.[Электронный ресурс].Режим доступа https://www.qt/io . Дата доступа 25.11.2023

[9] "Алгоритмы. Построение и анализ" Thomas H. Cormen, Charles E. Leiserson,

Ronald L. Rivest, Clifford Stein

[10] "Структуры данных и алгоритмы в C++" Robert Lafore.

[11] Сайт SiperComputer`s Blog [Электронный ресурс] .Режим доступа:https://chao-ji.github.io/jekyll/update/2018/07/19/BilinearResize.html . Дата доступа 25.11.2023

ПРИЛОЖЕНИЕ А
(обязательное)
Диаграмма классов

ПРИЛОЖЕНИЕ Б
(обязательное)
Схема метода *resized_image(Image *image,int new_height,int new_width)*

ПРИЛОЖЕНИЕ В
(обязательное)
Схема метода *ascii (Image * image, std::string file_name)*

ПРИЛОЖЕНИЕ Г
(обязательное)
Код программы

ПРИЛОЖЕНИЕ Д
(обязательное)
Ведомость документов