МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

КУРСОВАЯ РАБОТА

по дисциплине «Программирование»

Тема: Обработка BMP изображений

Студент гр. 0384 _____ Дзаппала Д.

Преподаватель _____ Шевская Н.В.

Санкт-Петербург

2021

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

Студент Дзаппала Даниэле

Группа 0384

Тема работы: Обработка BMP изображений

Исходные данные:

Общие сведения

- 24 бита на цвет
- без сжатия
- файл всегда соответствует формату BMP (но стоит помнить, что версий у формата несколько)
- обратите внимание на выравнивание; мусорные данные, если их необходимо дописать в файл для выравнивания, должны быть нулями.
- обратите внимание на порядок записи пикселей
- все поля стандартных BMP заголовков в выходном файле должны иметь те же значения что и во входном (разумеется кроме тех, которые должны быть изменены).

Программа должна реализовывать весь следующий функционал по обработке bmp-файла

1. Рисование правильного шестиугольника. Шестиугольник определяется:
    - **либо** координатами левого верхнего и правого нижнего угла квадрата, в который он вписан, **либо** координатами его центра и радиусом в который он списан
    - толщиной линий
    - цветом линий
    - шестиугольник может быть залит или нет
    - цветом которым залит шестиугольник, если пользователем выбран залитый
2. Копирование заданной области. Функционал определяется:
    - Координатами левого верхнего угла области-источника
    - Координатами правого нижнего угла области-источника
    - Координатами левого верхнего угла области-назначения
3. Заменяет все пиксели одного заданного цвета на другой цвет. Функционал определяется:
    - Цвет, который требуется заменить

- Цвет на который требуется заменить

4. Сделать рамку в виде узора. Рамка определяется:
    - Узором (должно быть несколько на выбор. Красивый узор можно получить используя фракталы)
    - Цветом
    - Шириной

Содержание пояснительной записки:

«Содержание»

«Введение»

«BMP-файл»

«Функции обработки изображения»

«Заключение»

«Список используемых источников»

Предполагаемый объем пояснительной записки:

Не менее 12 страниц.

Дата выдачи задания: 05.04.2021

Дата сдачи реферата: 29.05.2021

Дата защиты реферата: 31.05.2021

Студент _____ Дзаппала Д.

Преподаватель _____ Шевская Н.В.

АННОТАЦИЯ

Программа реализована как стандартная linux-утилита, где все команды прописываются как аргументы программы в терминале. Пользователь должен прописать опции, чтобы программа модифицировала изображение. Также, пользователь должен воспользоваться ключами, отвечающими за файлы (исходный файл и файл, куда будут сохраняться изменения (можно прописать тот же исходный)). Исходный файл должен либо находиться в той же директории, что и сама программа, либо можно прописать путь до файла. Если все ключи прописаны верно, и файл был найден, и программа смогла прочитать оттуда данные, то программа модифицирует изображение и сохраняет его в файл.

SUMMARY

The program is implemented as a standard linux utility, where all commands are written as program arguments in the terminal. The user must write the options for the program to modify the image. Also, the user must use the keys responsible for the files (the source file and the file where the changes will be saved (you can write the same source file)). The source file must either be in the same directory as the program itself, or you can write the path to the file. If all the keys are written correctly, and the file was found, and the program was able to read data from there, then the program modifies the image and saves it to the file.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель работы заключалась в том, чтобы научиться работать с bmp файлами, т.е. как правильно считывать информацию из бинарного файла, как модифицировать данные, и как сохранять их, соответственно. Также, нужно было реализовать программу в виде утилиты, чтобы можно было задавать опции прямо из терминала. Нужно было реализовать следующий функционал: рисование правильного шестиугольника, с выбором толщины линии, ее цвета, а так же будет ли залит шестиугольник; копирование заданной области; замена всех пикселей одного цвета на другой; рамка в виде узора. Для реализации CLI была использована библиотека getopt.h.

1. BMP-ФАЙЛ

1.1. Чтение bmp-файла

Чтобы модифицировать файл, его сначала нужно «прочитать». BMP файлы состоят из 3 блоков:

1. BITMAPFILEHEADER — 14-байтная структура, которая располагается в самом начале файла. Эта структура полезна, чтобы удостовериться в том, является ли файл BMP.
2. BITMAPINFO — эта структура имеет в зависимости от версии BMP файла свой размер. Самая важная информация лежит именно здесь.
3. Пиксели.

В работе были созданы две структуры: file_header и file_info_header. Структура file_info_header содержит в себе 11 полей на 40 байт памяти, это данные на 3-ю версию bmp файла. Также отдельно было создана структура с совмещенными данными уже описанных двух структур. Она используется только один раз в программе для проверки данных файла, который задает пользователь. Идет проверка на то, является ли файл «Битмаповским», версия файла, и сжат ли файл или нет.

Основа всей программы, это реализованный класс bmp_v3, содержащий в private области две переменные структуры (описанные выше), а также указатель на указатель на массив структур rgb. rgb, это структура, содержащая в себе три поля значения цветов. Был реализован конструктор класса, в аргументы которого подается указатель на конст. строку(файл, заданный пользователем). В конструкторе создается объект класса ifstream, который в себе и содержит данные файла. Если файл не открылся, то идет выход из программы. Если нет, то дальше с помощью метода read() класса ifstream, мы «читаем» данные в наши структуры. Читаем в порядке file_header, file_info_header. После, идет чтение пикселей, построчно.

1.2. Запись данных в файл

Чтобы записать данные в файл, была перегружена операция вывода для класса ostream. То есть, в main ф-ии инициализируется объект класса ofstream (файлом, в который надо сохранить), и в него с помощью метода write() записываются данные объекта класса bmp_v3. Данные записываются в том же порядке, что и при считывании, т.е. вначале записывается переменная file_header, после file_info_header, а в самом конце все пиксели. После чего программа завершает свою работу.

2. ФУНКЦИОНАЛ

2.1. Рисование правильного шестиугольника

Правильный шестиугольник можно определить с помощью окружности и квадрата. Для окружности нужны координаты центра окр. и его радиус. Квадрат определяется координатами верхнего левого угла квадрата и координатами нижнего правого угла квадрата. Также, шестиугольник определяется толщиной линии, цветом линии, шестиугольник может быть залит, и цвет заливки, в случае если был выбран залитый режим.

Метод **draw_hexagon_circ()** с помощью параметрической формулы нахождения окружности, находит 6 точек правильного шестиугольника, записывая их в массив структур типа point, поля корого есть координаты x и y. После чего с помощью метода **draw_line()** соединяются точки и получается правильный шестиугольник. Если пользователь решил залить фигуру, то вызывается метод **fill().**

В CLI, чтобы нарисовать шестиугольник нужно вызвать опцию —hexagon или коротко -X. Чтобы задать координаты центра окружности есть опция --circle_centre (- C), которая ожидает два параметра, координаты x и y. Для радиуса есть опция --radius (-l), которая требует один параметр. Чтобы задать квадрат, есть опции --top_coords(-t) и --bottom_coords(-b), обе ожидают два параметра: x и y. --line_thickness(короткий ключ -L), требует один параметр: толщину линии. --color(короткий ключ -c), требуется 3 параметра: красный цвет, зеленый цвет и синий цвет.(цвета вводить от 0 до 255) По умолчанию черный цвет. Можно использовать для задания цветов линии или для изначальных пикселей для функции --change_color (-A). --fill(короткий ключ -f), заливка шестиугольника, требует один параметр: 0 - если без заливки (по умолчанию), 1 - если залитая фигура. --fill_color(короткий ключ -F), выбор цвета заливки шестиугольника, требуется 3 параметра: красный цвет, зеленый цвет и синий цвет.(цвета вводить от 0 до 255) По умолчанию черный цвет.

2.2. Копирование области изображения

Определяется: Координатами левого верхнего угла области-источника, Координатами правого нижнего угла области-источника, Координатами левого верхнего угла области-назначения. Создается новый двумерный массив, куда будут записываться пиксели заданной области, построчно записываем данные, если все проверки проходят, благополучно вставляем скопированную область в новую область.

В CLI созданы следующие опции:

--copy_region (короткий ключ -R), не требует параметров. Функция копирует область изображения и переставляет в другое место. Чтобы использовать функцию нужно воспользоваться опциями --top_coords (-t) и --bottom_coords (-b) для задания копируемой области. С помощью опции --region_coords (-r) нужно задать координаты левого верхнего угла области назначения. --region_coords (короткий ключ -r), требует два параметра: координаты (X, Y) левого верхнего угла области назначеия для функции копирования области изображения.

2.3. Изменение цветов пикселей

Определяется: цвет, который требуется заменить; цвет, на который требуется заменить. Функция **exact_changement()** сравнивает цвета, и если есть совпадение, перезаписывает пиксели тем цветом, который пользователь задал.

Для CLI созданы следующие опции:

--change_color(короткий ключ -A), опция не требует параметров. Функция меняет все пиксели заданного цвета на пиксели нового цвета. Чтобы задать цвет, который вы хотите заменить, воспользуйтесь опцией --color (-c). Для задания нового цвета используйте опцию --new_color (-N). --new_color (короткий ключ -N), опции требуется три параметра: красный цвет, зеленый цвет и синий цвет.(цвета вводить от 0 до 255) По умолчанию черный цвет.

2.4. Рисование рамки узора

Были создана функция, в которую подается какой узор нарисовать, который определяет пользователь. Были созданы узоры: обычная рамка, прямоугольный зиг-заг, треугольники, половинчатая синусоида, окружности.

ЗАКЛЮЧЕНИЕ

Задача выполнена успешно. Все задачи выполнены. Была освоена структура BMP файлов и то, как их модифицировать. А также, как делать CLI.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Лекции по программированию;

Стивен Прата - «Язык программирования С++»;

Интернет;

Видео на You-Tube.