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

КУРСОВАЯ РАБОТА

по дисциплине «Программирование»

Тема: Обработка PNG файла

Студент гр. 3341 _____ Мальцев К.Л.

Преподаватель _____ Глазунов С.А.

Санкт-Петербург

2024

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

Студент Мальцев К.Л.

Группа 3341

Вариант 5.17

Программа обязательно должна иметь CLI (опционально дополнительное использование GUI). Более подробно тут: http://se.moevm.info/doku.php/courses:programming:rules_extra_kurs

Программа должна реализовывать весь следующий функционал по обработке png-файла

Общие сведения

Формат картинки PNG (рекомендуем использовать библиотеку libpng)

без сжатия

файл может не соответствовать формату PNG, т.е. необходимо проверка на PNG формат. Если файл не соответствует формату PNG, то программа должна завершиться с соответствующей ошибкой.

обратите внимание на выравнивание; мусорные данные, если их необходимо дописать в файл для выравнивания, должны быть нулями.

все поля стандартных PNG заголовков в выходном файле должны иметь те же значения что и во входном (разумеется кроме тех, которые должны быть изменены).

Программа должна иметь следующую функции по обработке изображений:

Рисование окружности. Флаг для выполнения данной операции: `--circle`. Окружность определяется:

координатами ее центра и радиусом. Флаги `--center` и `--radius`. Значение флаг `--center` задаётся в формате `x.y`, где x – координата по оси x, y – координата по оси y. Флаг `--radius` На вход принимает число больше 0

толщиной линии окружности. Флаг `--thickness`. На вход принимает число больше 0

цветом линии окружности. Флаг `--color` (цвет задаётся строкой `rrr.ggg.bbb`, где rrr/ggg/bbb – числа, задающие цветовую компоненту. пример `--color 255.0.0` задаёт красный цвет)

окружность может быть залитой или нет. Флаг `--fill`. Работает как бинарное значение: флага нет – false , флаг есть – true.

цветом которым залита сама окружность, если пользователем выбрана залитая окружность. Флаг `--fill_color` (работает аналогично флагу `--color`)

Фильтр rgb-компонент. Флаг для выполнения данной операции: `--rgbfilter`. Этот инструмент должен позволять для всего изображения либо установить в диапазоне от 0 до 255 значение заданной компоненты. Функционал определяется

Какую компоненту требуется изменить. Флаг `--component_name`. Возможные значения `red`, `green` и `blue`.

В какой значение ее требуется изменить. Флаг `--component_value`. Принимает значение в виде числа от 0 до 255

Разделяет изображение на N*M частей. Флаг для выполнения данной операции: `--split`. Реализация: провести линии заданной толщины. Функционал определяется:

Количество частей по “оси” Y. Флаг `--number_x`. На вход принимает число больше 1

Количество частей по “оси” X. Флаг `--number_y`. На вход принимает число больше 1

Толщина линии. Флаг `--thickness`. На вход принимает число больше 0

Цвет линии. Флаг `--color` (цвет задаётся строкой `rrr.ggg.bbb`, где rrr/ggg/bbb – числа, задающие цветовую компоненту. пример `--color 255.0.0` задаёт красный цвет)

Рисование квадрата с диагоналями. Флаг для выполнения данной операции: `--squared_lines`. Квадрат определяется:

Координатами левого верхнего угла. Флаг `--left_up`, значение задаётся в формате `left.up`, где left – координата по x, up – координата по y

Размером стороны. Флаг `--side_size`. На вход принимает число больше 0

Толщиной линий. Флаг `--thickness`. На вход принимает число больше 0

Цветом линий. Флаг `--color` (цвет задаётся строкой `rrr.ggg.bbb`, где rrr/ggg/bbb – числа, задающие цветовую компоненту. пример `--color 255.0.0` задаёт красный цвет)

Может быть залит или нет (диагонали располагаются “поверх” заливки). Флаг `--fill`. Работает как бинарное значение: флага нет – false , флаг есть – true.

Цветом которым он залит, если пользователем выбран залитый. Флаг `--fill_color` (работает аналогично флагу `--color`)

Каждую подзадачу следует вынести в отдельную функцию, функции сгруппировать в несколько файлов (например, функции обработки текста в один, функции ввода/вывода в другой). Сборка должна осуществляться при помощи make и Makefile или другой системы сборки

Дата выдачи задания: 18.03.2024

Дата сдачи реферата: 13.05.2024

Дата защиты реферата: 15.05.2024

Студент		Мальцев Н.И.
Преподаватель		Глазунов С.А.

АННОТАЦИЯ

В данной курсовой работе была реализована программа, обрабатывающая PNG изображения, не имеющие сжатия. Программа проверяет тип изображения, его версию, при соответствии требованиям в дальнейшем обрабатывает его и подаёт на выход изменённую копию изображения. Взаимодействие с программой осуществляется с помощью CLI (интерфейс командной строки).

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью данной работы является создание программы на языке C++ для обработки PNG изображений.

Для достижения поставленной цели потребовалось решить ряд задач:

- изучить, как устроены PNG файлы, что они в себе содержат;

- научиться распознавать PNG файлы среди прочих и проверять их прочие характеристики;

- научиться считывать и записывать PNG изображения;

- разработать функцию рисования треугольника на изображении, его заливки;

- разработать функцию поиска наибольшего прямоугольника заданного цвета на изображении и его перекрашивания;

- разработать функцию создания коллажа из исходного изображения по заданным количествам повторений изображения по оси x и по оси y;

- изучить библиотеку *getopt.h*;

- научиться работать с аргументами командной строки, длинными и короткими флагами;

- создать *Makefile* для сборки программы;

- протестировать разработанную программу.

ХОД РАБОТЫ

В директории src хранятся исходники, в include — хедеры, в libs — использованные библиотеки. Для запуска утилиты требуется собрать её через makefile командами make build_libs и затем — make.

Использованная библиотека — ImageEditor (ссылка на ресурс: https://github.com/KirillMaltsev3341/ImageEditor). В ней представлен базовый функционал для обработки png и bmp файлов. В директории docs находится doxygen файл, с помощью которого можно сгенирить подробную документацию о функционале данной библиотеки.

В файле main.cpp вызывается статический метод ShowAuthorInfo класса Output для вывода информации об авторе курсовой. Далее создается объект класса Handler для обработки флагов. Метод getFlags записывает входные флаги во внутренний словарь объекта handler. Метод handleFlags обрабатывает данный набор флагов и вызывает соответствующую do-функцию, каждая из которых соответствует заданию курсовой.

Каждая из do-функций обрабатывает словарь флагов handler-а и переданные аргументы при помощи набора функций из namespace psr (парсер). Разберем работу одной из do-функций (например doCircle). Сначала создаются переменные, в которые считаются аргументы флагов при помощи парсера (color, thickness, input_file_name, …). Далее после обработки аргументов парсером создается объект класса ie::ImagePNG. С помощью метода readImageFromFile в него записывается информация о входном изображении. Далее вызывается функция drawCircle с аргументами, которые обработал парсер. После чего новое изображение записывается в выходной файл с помощью метода writeImageToFile.

Работа остальных do-функций аналогична.

ЗАКЛЮЧЕНИЕ

Разработана программа на языке программирования C++, обрабатывающая PNG изображения и имеющая CLI. В ходе выполнения работы было изучено устройство PNG файлов; изучены методы считывание и записи файлов; получены навыки обработки изображений; изучена библиотека *getopt.h;* изучена работа с аргументами командной строки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1.https://ru.wikipedia.org/wiki/ PNG - структура файла PNG;

2.https://se.moevm.info/lib/exe/fetch.php/courses:programming:programming_cw_metoda_2nd_course_last_ver.pdf.pdf - методические материалы для написания курсовой работы

3.https://habr.com/ru/articles/55665/ - принцип работы getopt_long

4.https://ru.wikibooks.org/wiki/Реализации_алгоритмов/Алгоритм_Брезенхэма - алгоритм Брезенхэма.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Использованная библиотека: https://github.com/KirillMaltsev3341/ImageEditor

Код курсовой работы: https://github.com/KirillMaltsev3341/term2_cw

ПРИЛОЖЕНИЕ Б
ТЕСТИРОВАНИЕ

Test 1:

```
./cw --squared_lines --left_up 420.200 --side_size 200 --thickness 5 --
color 255.0.0 --fill --fill_color 0.0.255 input.png
```

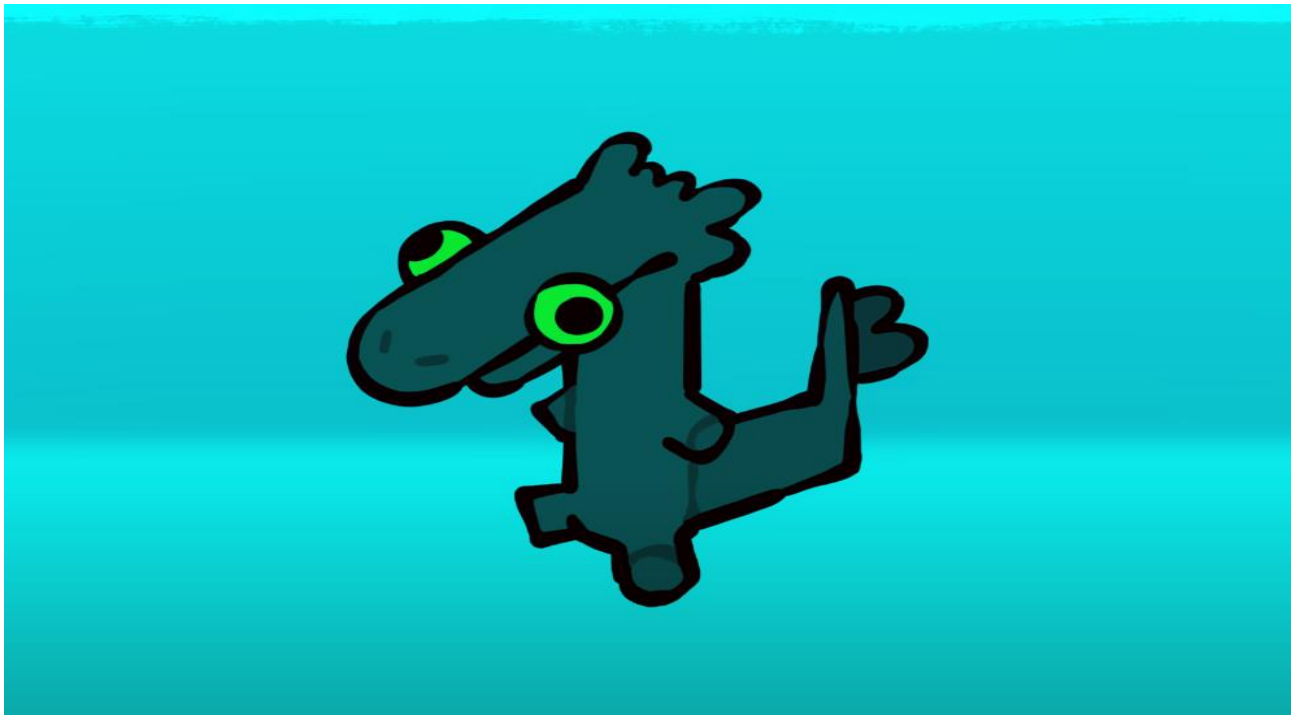
Результат работы программы:


MACプラス
フローラルの専門店

Test 2:

```
./cw --rgbfilter --component_name red --component_value 10 input.png
```

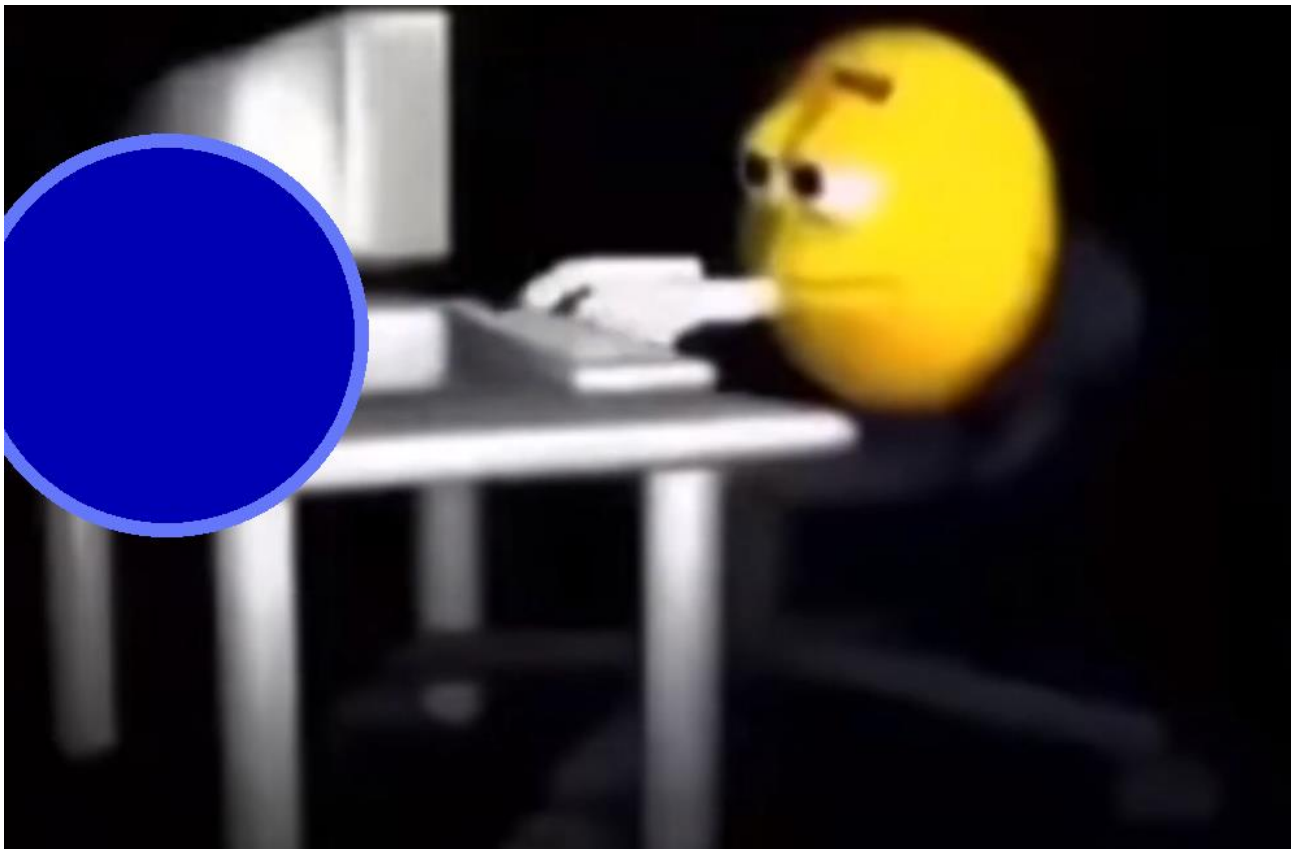
Результат работы программы:



Test 3:

```
./cw --circle  --center 125.254 --thickness 10 --color 101.119.250 --
fill_color 0.0.179 --fill  --radius 150 --input ./input.png --output
./output.png
```

Результат работы программы:



Test 4:

```
./cw --split --output ./output.png --number_y 10 --input ./input.png --
thickness 2 --number_x 20 --color 3.25.10
```

Результат работы программы: