

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

КУРСОВАЯ РАБОТА
по дисциплине «Программирование»
Тема: Обработка изображений.

Студент гр. 0303

Денежный А.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент: Денежный А.А.

Группа 0303

Тема работы: Обработка изображений

Исходные данные:

Вариант 12

Программа **должна** иметь CLI или GUI.

Общие сведения

- **Формат картинки PNG (рекомендуем использовать библиотеку `libpng`)**
- файл всегда соответствует формату PNG
- 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 реализовывать следующий функционал по обработке PNG-файла:

1. Рисование квадрата. Квадрат определяется:
 - Координатами левого верхнего угла
 - Размером стороны
 - Толщиной линий
 - Цветом линий
 - Может быть залит или нет
 - Цветом которым он залит, если пользователем выбран залитый

2. Поменять местами 4 куса области. Выбранная пользователем прямоугольная область делится на 4 части и эти части меняются местами. Функционал определяется:
 - Координатами левого верхнего угла области
 - Координатами правого нижнего угла области
 - Способом обмена частей: “по кругу”, по диагонали
3. Находит самый часто встречаемый цвет и заменяет его на другой заданный цвет. Функционал определяется Цветом, в который надо перекрасить самый часто встречаемый цвет.
4. Инверсия цвета в заданной области. Функционал определяется
 - Координатами левого верхнего угла области
 - Координатами правого нижнего угла области

Содержание пояснительной записки:

Разделы: Введение, реализация структур, функции чтения и записи изображения, функции по обработке изображения, создание CLI, запуск программы.

Предполагаемый объем пояснительной записки:

Не менее 21 страниц.

Дата выдачи задания: 05.04.2021

Дата сдачи реферата: 30.05.2021

Дата защиты реферата: 01.06.2021

Студент		Денежный А.А.
Преподаватель		Чайка К. В.

АННОТАЦИЯ

Курсовая работа заключалась в создании проекта для чтения, обработки изображения формата png в соответствии с параметрами, заданными пользователем, и записи его в файл. При разработке программы были реализованы структуры Png, Color, ColorCount и Coordinates, а также созданы функции для обработки изображения. Результатом работы стали программа, в которой был реализован весь необходимый функционал и интерфейс командной строки (CLI) для более удобного взаимодействия с пользователем.

СОДЕРЖАНИЕ

Введение	6
1. Реализация структур	7
2. Функции чтения и записи изображения	8
3. Функции по обработке изображения	9,10
4. Создание CLI	10
5. Запуск программы	11,12
Заключение	13
Список использованных источников	14
Приложение А. Примеры работы программы	15

ВВЕДЕНИЕ

Целью данной работы является создание консольного приложения, способного обработать входной файл формата PNG в соответствии с подзадачами и дополнительными условиями.

Для достижения поставленной цели требуется решить следующие задачи:

1. Считывание изображения в структуру *Png*.
2. Реализация интерфейса в командной строке в виде CLI для запроса действий от пользователя.
3. Реализация подзадач:
 - Нарисовать квадрат.
 - Поменять местами 4 куска области.
 - Найти самый часто встречаемый цвет и заменять его на другой заданный цвет.
 - Инверсия цвета в заданной области.

1. РЕАЛИЗАЦИЯ СТРУКТУР.

В ходе разработки программы было реализовано 4 структуры:

1.1 *struct Png.*

Данная структура содержит всю необходимую информацию об изображении, а именно:

Int height, width - целочисленные поля, содержащие высоту и ширину изображения в пикселях соответственно.

png_byte color_type - поле, хранящее тип цветовой модели (RGBA, RGB и др.).

png_byte bit_depth - поле, хранящее информацию о глубине цвета изображения.

png_bytep *row_pointers - поле, являющееся двумерным массивом строк пикселей изображения. Основные преобразования изображения происходят посредством различных манипуляций над данным массивом.

1.2 *struct Color.*

Элемент данной структуры является характеристикой какого-либо цвета:

char name [] - массив символов, в котором хранится название цвета, удобное для восприятия человеком. Например, red, blue, green и тд.

unsigned char r, g, b - числовые поля, которые хранят в себе информацию о красной, синей и зеленой составляющих конкретного цвета.

1.3 *struct CountColor.*

Данная структура необходима в реализации одной из функций программы.

long long int num, count - поля, хранящие большие целочисленные значения.

1.4 *struct Coordinates.*

Элемент данной структуры является аналогом точки на плоскости, то есть хранит в себе координаты конкретной точки на плоскости.

int x,y - горизонтальная и вертикальная составляющие точки соответственно.

2. Функции чтения и записи изображения.

2.1 Функция *readPngFile*.

Данная функция работает с некоторым файлом, который передаёт пользователь, и, если выполнены все условия и исключены все ошибки, считывает информацию о файле в объект структуры *Png*.

2.2 Функция *writePngFile*.

Данная функция работает объектом структуры *Png*: она создаёт новый файл типа *png* и заносит туда информацию о изображении *image*.

3. Функции по обработке изображения.

3.1 Функция *drawSquare(struct Png* image, struct Coordinates leftAngle, int side, int depth, struct Color* lineColor, struct Color* fillColor).*

Данная функция принимает указатель на структуру, хранящую в себе информацию об изображении, координаты точки на плоскости, в которых будет расположен левый верхний угол квадрата, длину стороны квадрата, толщину его линий, их цвет и цвет заливки квадрата, если это необходимо.

Функция производит проверку переданных значений и, если они корректны, функция вызывает 4 раза функцию рисования прямоугольника цвета, который был передан в качестве цвета линий квадрата, а затем, если это необходимо, функцию заполнения квадрата цветом, переданным для заполнения.

Вспомогательная функция ***void drawRect(struct Png* image, int x, int y, int height, int width, struct Color* color).***

Рисует закрашенный прямоугольник на изображении путем замены цвета пикселей в заданном диапазоне на переданный ей цвет.

Вспомогательная функция ***void fillSquare(struct Png* image, int x_begin, int y_begin, int x_end, int y_end, struct Color* color).***

Данная функция предназначена для заполнения области внутри квадрата переданным ей цветом, если это было указано пользователем, иначе данная функция вызвана не будет.

Если координаты переданные функции *drawSquare* окажутся некорректны, она сообщит об этом пользователю и завершит работу программы.

3.2 Функция *void swap(struct Png* image, struct Coordinates leftAngle, struct Coordinates rightAngle, char* mode).*

Данная функция производит смену частей картинки в определённой области в зависимости от режима, выбранного пользователем. Функция принимает указатель на структуру, хранящую в себе информацию об

изображении, координаты точек на плоскости, отвечающих за левый верхний и правый нижний углы выбранной пользователем области, а также режим, которым будет определена работа функции - смена частей изображения по кругу или по диагонали. Функция создает копию полученного изображения и в зависимости от координат производит обмен пикселей исходного изображения на его копию.

3.3 Функция void changeColor(struct Png* image, struct Color* new_color).

Данная функция принимает указатель на структуру, хранящую в себе информацию об изображении и цвет, на который будет заменен самый часто встречающийся цвет на исходном изображении. В функции происходит подсчет всех пикселей каждого цвета, а затем замена самого часто встречающегося цвета на новый.

3.4 Функция void invertColors(struct Png* image, struct Coordinates leftAngle, struct Coordinates rightAngle).

Данная функция принимает указатель на структуру, хранящую в себе информацию об изображении и координаты точек левого верхнего и правого нижнего углов области, выбранной пользователем для инвертирования цветов. Если пользователь не передаст никакие координаты, то изображение будет инвертировано целиком.

4. СОЗДАНИЕ CLI.

Для создания CLI была использована функция стандартной библиотеки языка C - *getopt_long()*.

5.Запуск программы.

Сборка программы осуществляется в директории с проектом при помощи следующей инструкции для терминала: **make**

ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ

По умолчанию последним аргументом программа принимает имя *PNG* файла, имя файла *PNG* после обработки - *out.png*.

Для того чтобы увидеть справку, нужно запустить программу с одним из следующих ключей: *--help / -h / -?*

Пример: *./cw2 -h*

Для того чтобы вручную задать имя файла после обработки необходимо использовать ключ *--out / -o*

Пример: *./cw2 <имя_файла_для_обработки> -o
<имя_файла_после_обработки>*

Координаты левого верхнего угла задаются ключом *--start / -s*, координаты принимаются в виде: *x,y*

Координаты правого нижнего угла задаются ключом *--end / -e*, координаты принимаются в виде: *x,y*

Пример: *./cw2 --start x1,y1 --end x2,y2 <имя_файла_для_обработки>*

Для того чтобы нарисовать квадрат, необходимо использовать ключ *--draw-square / -D*, который принимает обязательные параметры - длину стороны и толщину линий квадрата в виде: *сторона,толщина*

Для того чтобы указать, каким цветом будет залит квадрат, необходимо воспользоваться флагом *--fill-color / -f* и передать ему аргумент - название цвета

Для того чтобы указать цвет линий квадрата, необходимо воспользоваться ключом *--line-color / -l* и передать обязательный параметр - имя цвета

Примеры: *./cw2 -D side,depth --start x,y -f <имя_цвета> -l <имя_цвета>
<имя_файла_для_обработки>*

*./cw2 -D side,depth --start x,y -l <имя_цвета>
<имя_файла_для_обработки>*

Для того чтобы поменять местами 4 куска области, необходимо воспользоваться ключом *--swap-parts / -S*, который принимает обязательный аргумент - режим работы: *circle* или *diagonal*

Пример: *./cw2 -S diagonal --start x1,y1 --end x2,y2
<имя_файла_для_обработки>*

Для того чтобы найти и заменить самый часто встречаемый цвет, необходимо воспользоваться ключом **--change-most-frequent-color / -C**, который принимает обязательный аргумент - цвет для замены

Пример: `./cw2 -C <имя_цвета> <имя_файла_для_обработки>`

Для того чтобы инвертировать цвета изображения, необходимо воспользоваться ключом **--invert-colors / -I**. По умолчанию инвертируются все цвета на картинке

Примеры: `./cw2 -I --start x1,y1 --end x2,y2 <имя_файла_для_обработки>`

`./cw2 -I <имя_файла_для_обработки>`

Таблица имен цветов:

red	>> красный
blue	>> синий
tgreen	>> зеленый
black	>> черный
white	>> белый
orange	>> оранжевый
violet	>> фиолетовый
gray	>> серый
lightblue	>> голубой
yellow	>> желтый

ЗАКЛЮЧЕНИЕ

В результате работы было реализовано считывание PNG-файла в структуру и созданы различные функции для обработки пикселей, интерфейс в виде CLI для взаимодействия с пользователем и получена программа, работающая на Linux.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Получение основных условий курсовой работы

http://se.moevm.info/doku.php/courses:programming:rules_extra_kurs/

Дата обращения: 12.05.2021

2. Основная информация о формате PNG <https://biz-iskun.ru/format-png-chto-eto.html>

Дата обращения: 12.05.2021

4. Б.В. Керниган, Д.М. Ричи. Язык программирования С. 3-е издание.

5. Пример реализации CLI <https://habr.com/ru/post/150950/>

Дата обращения: 25.05.2021

6. Основная информация о CLI <https://ru.wikipedia.org/wiki/C%2B%2B/CLI>

ПРИЛОЖЕНИЕ А

Примеры работы программы.

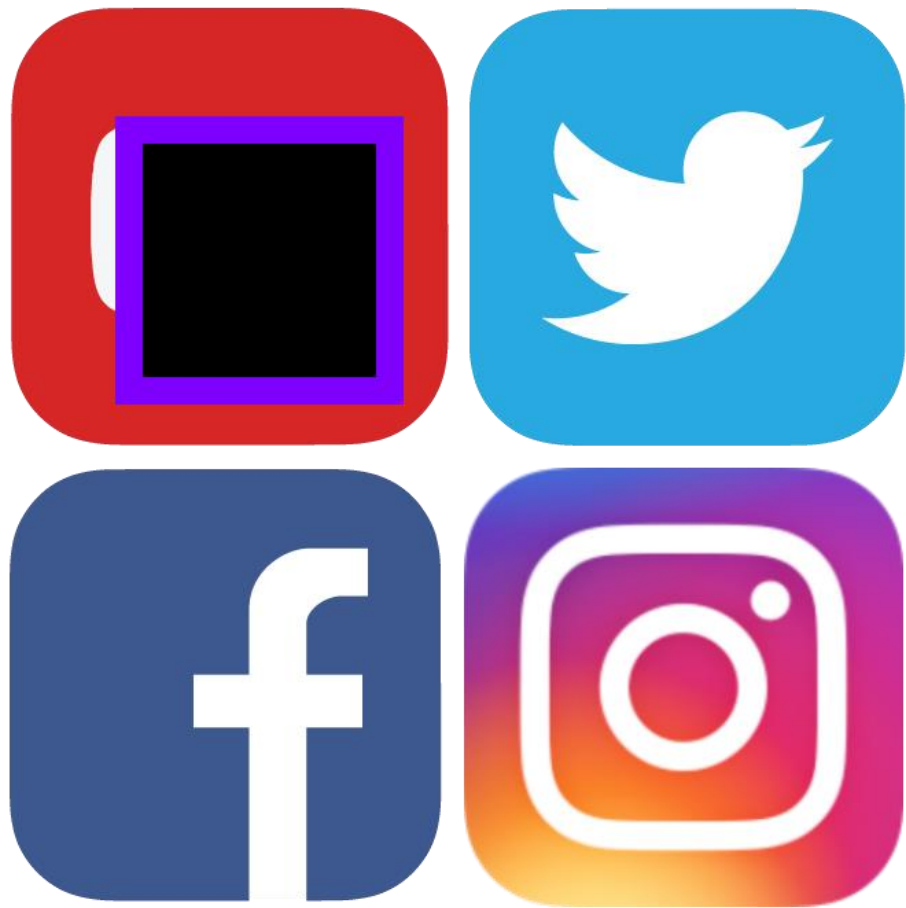
Инвертирование цветов изображения:



Замена самого часто встречающегося цвета:



Рисование квадрата:





Смена местами 4 частей изображения:





