Assignment 5
Color Blindness Simulator

By Jiancheng (Jason) Xiong
CSE 13S Spring 2023

Due May 28th 2023, 11:59 PM

1 Purpose

The purpose of this assignment is to create a program that takes an input BMP image and alters the RGB values in a way to simulate the view of a dichromat, then to output the image in a new file. The program will use unbuffered reads and writes to take an input image and write it to a new file. The program will also parse command line arguments that specify

- The input file/image
- The outfile file/image
- A help message if requested

2 Using the Program

In order to use the program, compile and run the file colorb.c. In order to run the program, run ./colorb and specify the commands in which you want to run. The available options are;

-i (inputfile), specifying the input image on which the program will run on.

-o (outputfile), specifying the output file in which the image will be stored

-h; displays a help message detailing program usage.

3 Program Design

The main program will be in the file colorb.c, while each of the separate data structure and BMP manipulation files will be in;

- Io.c for the reading and writing of various uint sizes, includes functions that open and close files unbuffered, along with header file io.h
- Bmp.c for the manipulation of the BMP image format; includes functions that read and write bmp image files, along with header file bmp.h

- Makefile for the compilation and joining of the separate files

4 Data Structures

There are various implemented data structures, being;

The buffer struct is used to facilitate the reading and writing of files. It stores the next available position for writing and reading, as well as storing the max size of the buffer and a pointer to its position.

The color struct is used to save the RGB values of each pixel of an image. Each red, green, and blue value is stored in a variable.

The BMP struct used to store width, height, an array of pixels, as well as colors for those pixels. The height and width of the image is used to make calculations later on, while the 2 dimensional array stores the actual pixels and their colors.

5 Algorithms

The main function:

- Parse through command line arguments
- If input or output file command line arguments are not found or incomplete, print an error message and exit program
- Check if help message was requested, if so, print help message and exit program
- Use BMP create to read the image and create a file on which we will work on
- Use BMP reduce to alter the RGB values of the image to simulate the vision of a Dichromat
- Use BMP write to write the BMP to the output file
- Free the BMP image
- Close the files for read and write

BMP functions:

- **Read**
    - Parse through the data in the input image and store the information to relevant variables, as we know the ordering and size of the information in the BMP file.

- **Write**
    - Create a file and write the relevant information with specific sizes and ordering in order to create a suitable BMP file

IO functions:

- **Read open**
    - Open a file and create a buffer, setting the positioning of off_set and num_remaining to 0, then return the pointer to that buffer
- **Read close**
    - Close the file, free the buffer, and set its pointer to null
- **Write open**
    - Open a file in write mode and create a buffer, setting the positioning of off_set and num_remaining to 0, then return the pointer to that buffer
- **Write close**
    - Close the file, free the buffer, and set its pointer to null
- **Read uint8_t**
    - Check if the buffer has any remaining bytes, and if there are none remaining, read a chunk from the file descriptor into the buffer array.
    - Update the number of remaining bytes and the offset of the bytes
    - Return true
- **Read uint16_t**
    - Call uint8_t, left shift it by 8, then or it with a second uint8_t
- **Read uint32_t**
    - Call uint16_t, left shift it by 16, then or it with a second uint16_t
- **Write uint8_t**
    - Check if buffer's size has been reached, if so flush the buffer
    - Write the data in the buffer to the file descriptor
    - Offset is reset to 0, and x is set to the current offset position
- **Read uint16_t**
    - Call uint8_t, right shift it by 8, then and it with a second uint8_t
- **Read uint32_t**
    - Call uint16_t, right shift it by 16, then and it with a second uint16_t

6 Function Descriptions

Main function: takes its arguments from the command line, using getopt to know whether enough options have been given to run. Takes one (command line) input and outputs the altered image in a new image.

All read u_ints: takes an input buffer and *x and saves what is read to x

All write u_ints: takes an input buffer and x and writes x to the buffer

Write open: takes an input filename and opens it in writing mode

Write close: takes an input filename and closes it, freeing the allocated data, and setting the pointer to null.

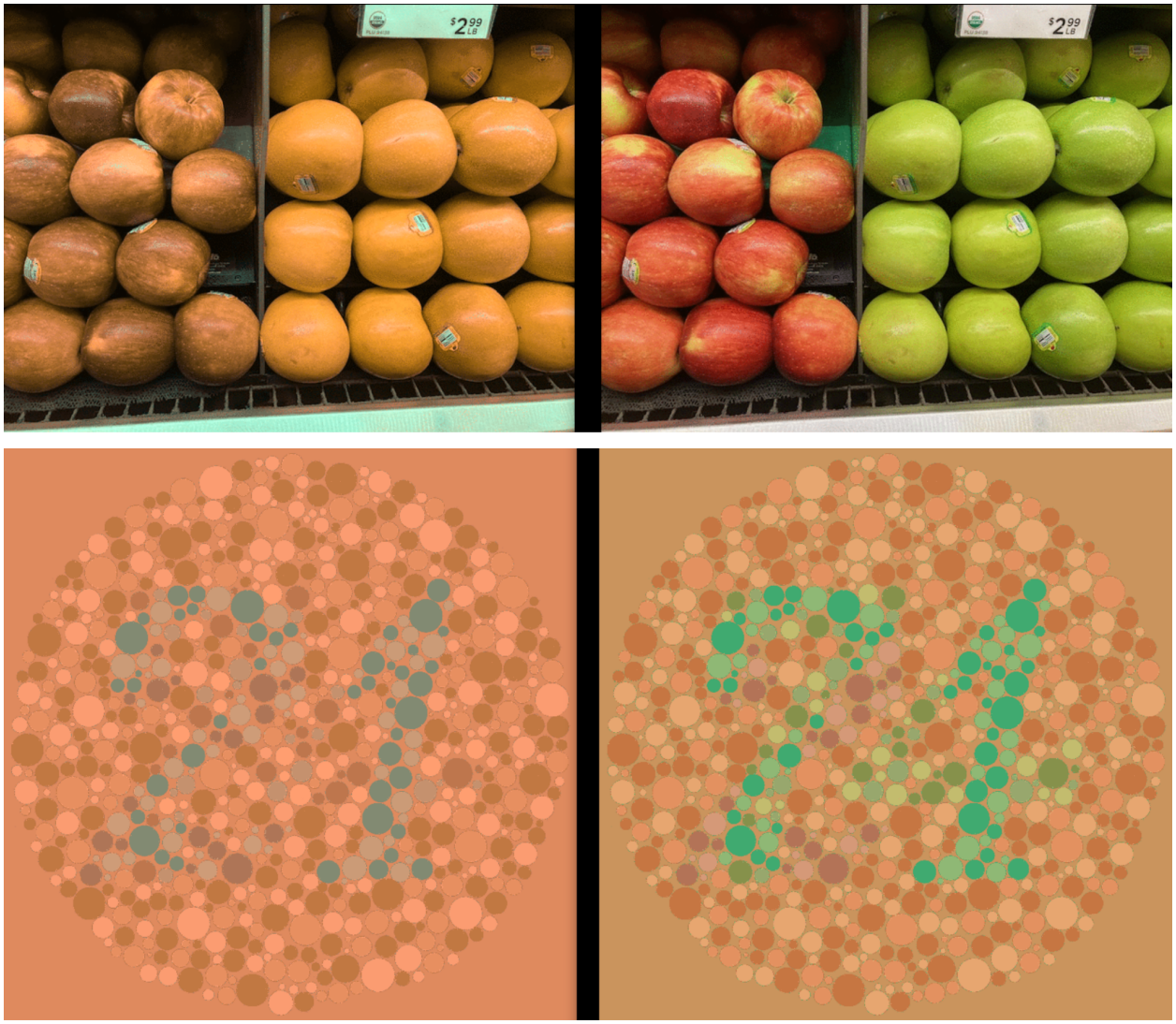
Read open: takes an input filename and opens it in reading mode

Read close: takes an input filename and closes it, freeing the allocated data, and setting the pointer to null.

7 Results

While writing this program I learned how to use unbuffered commands in order to save data while writing and reading in order to write efficient programs. I have also learned how Windows BMP files are formatted and read, as well as how we can alter characteristics of them by checking the data values at predetermined indices in the data. While writing my io read and write functions I have learned how to use the buffer in order to let the program know what it needs to write to the file.

Pictures of the program running:


$2.99 LB
$2.99 LB

8 References

"Getopt() Function in C to Parse Command Line Arguments." *GeeksforGeeks*, 10 Sept. 2018, www.geeksforgeeks.org/getopt-function-in-c-to-parse-command-line-arguments/.

Open(2) - Linux Manual Page, 27 Aug. 2021, man7.org/linux/man-pages/man2/open.2.html.

"BMP File Format." Wikipedia, 16 May 2023, en.wikipedia.org/wiki/BMP_file_format.