Programming Assignment 1

Due Date: Please see the Canvas assignment entry

This is an individual assignment.

Submission

- Submit your source code in a zip file on Canvas.
- Submit a PDF report on Crowdmark. You will receive a link by emails.
    - If you did not receive the Crowdmark assessment email, please let me know.

Use your student number to name the zip and PDF files.

For example, `301XXXX.zip` and `301XXXX.pdf`

Project Description

Write a program with GUI allowing users to open a BMP file and display the metadata and the BMP image.

Please use this page as a reference for BMP file format:

https://www.ece.ualberta.ca/~elliott/ee552/studentAppNotes/2003_w/misc/bmp_file_format/bmp_file_format.htm

You cannot use any library to parse the BMP file. You will need to write your own parser. To do that, you need to first read the BMP file into an array of bytes, and interpret the meaning of the bytes.

You can assume the BMP files are not using RLE compression.

For example, the following code reads the BMP file into an array of bytes:

```
with open("BIOS.bmp", "rb") as f:
    bmp_bytes = f.read()
```

The `“rb”` option here is to interpret the file into byte stream.

You will need to do the following things to parse the BMP file:

- parse the header for important metadata
- parse the bit map for the pixel data

The following code is an example of parsing the file size metadata from the header:

```
  def _get_file_size(self, bmp_bytes):
      return int.from_bytes(bmp_bytes[2:6], 'little')
```

BMP header uses little endian to encode their header metadata. You need to locate the location of those bytes that represent the file size. You can find their location from the referenced file format as shown below:

Name	Size	Offset
Header	14 bytes	
Signature	2 bytes	0000h
FileSize	4 bytes	0002h
reserved	4 bytes	0006h
DataOffset	4 bytes	000Ah

`0002h` represents the starting point of the bytes that represent the file size. `0002h` is in hexadecimal format.

You only need the following important metadata:

- File size
- Image width
  - Be careful about the padding.
  - Read the **Additional Info** from the referenced page above.
- Image height
  - Scan lines are from bottom to top.
  - Read the **Additional Info** from the referenced page above.
- Bits per pixel
  - Note that for bits per pixel that are 1, 4, 8 bits, the value represent an index of the color table and the actual color values are stored in the color table.
  - You only need to focus on bits per pixel that are 1, 4, 8, and 24 bits.

Display the above four metadata in your GUI.

You need to locate where is the bit map, where bit map contains the bytes for the pixel data (the actual colors or color table indexes). Parse the pixel data and display the BMP image on your GUI.

Pay attention to the `Bits Per Pixel Field` from the referenced file format. Depending on the number of bits per pixel, you need to parse each byte differently. Remember one byte consists of 8 bits.

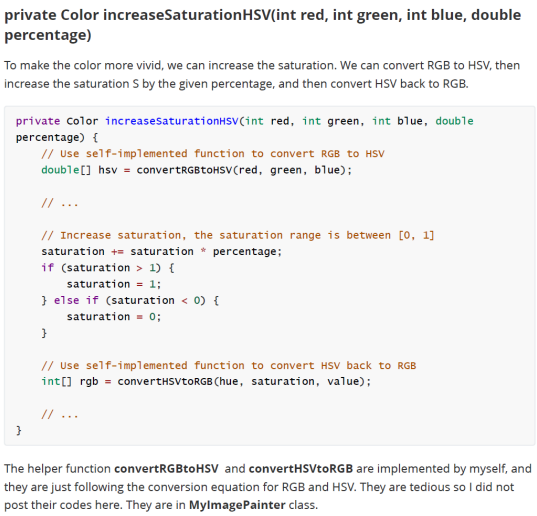
Your Program Requirements

- A reasonable GUI design
- Buttons to open a BMP file
- Should check whether file type is valid
  - Note that simply checking the extension name is not sufficient
- Display the metadata mentioned above
- Display the BMP image in a reasonable size
- Your program should not crash in any circumstances
- Implement the following features
  - User can adjust the brightness of the image using a slider from 0% to 100%
  - User can scale the image using a slider from 0% to 100%
    - You cannot use any library to rescale the image

        - You must find a reasonable way to aggregate the pixels
    - Provide three buttons for R, G, and B. By default, these buttons are enabled and images are displayed using all three RGB channels. User can click on the button to toggle enabled/disabled each corresponding RBG channel
- Ensure that you design your code and UI properly to allow future extension. We will build more features, such as compression, based on the program you built in this assignment

Report Requirements

- A title page with the class name, assignment title, your name, student ID, and school email.
- A section named `Program Screenshots` for your screenshots
    - A screenshot of your program GUI displaying the given sample BMP images
    - A series of screenshots showing that you have implemented all the requirements
- A section named `Essential Code Explanation`
    - Screenshot your essential functions and briefly explain what they did
    - Here is one example:



private Color increaseSaturationHSV(int red, int green, int blue, double percentage)

To make the color more vivid, we can increase the saturation. We can convert RGB to HSV, then increase the saturation S by the given percentage, and then convert HSV back to RGB.

```
private Color increaseSaturationHSV(int red, int green, int blue, double
percentage) {
    // Use self-implemented function to convert RGB to HSV
    double[] hsv = convertRGBtoHSV(red, green, blue);

    // ...

    // Increase saturation, the saturation range is between [0, 1]
    saturation += saturation * percentage;
    if (saturation > 1) {
        saturation = 1;
    } else if (saturation < 0) {
        saturation = 0;
    }

    // Use self-implemented function to convert HSV back to RGB
    int[] rgb = convertHSVtoRGB(hue, saturation, value);

    // ...
}
```

The helper function **convertRGBtoHSV** and **convertHSVtoRGB** are implemented by myself, and they are just following the conversion equation for RGB and HSV. They are tedious so I did not post their codes here. They are in **MyImagePainter** class.

    As can be seen from the example, you do not need to give all the details. Just help us understand what your code does is sufficient.

Sample Test Files

Please download the sample BMP files from Canvas.

Grading

You will receive a zero mark if you do not submit a report.

TAs will grade your program based on your report. Ensure that you show all the following items on your report.

- Reasonable UI: 2pt
- Report format and readability: 4pt
    - Refer to the **Report Requirements**

- A button to browse files and open a file: 2pt
- Should check whether file type is valid: 2pt
- Display the selected BMP file: 2pt
- Display the metadata: 4pt
- Implement features
  - slider to adjust brightness: 2pt
  - slider to scale image: 4pt
    - If not aggregating the pixels by yourself: (-2pt)
  - Buttons to toggle RGB channels: 2pt
- If the **Program Requirements** mentioned above but not listed here are not satisfied, -2pt for each missing requirement.