README

How to Use the Program:

Once started, the program will and the first thing it will ask it which program you want to run the Image Processing System.

-text : To use the interactive text mode.

Interactive text mode is the one initially made that causes you to input commands individually. The keywords to run specific commands:

I. **brighten <increment> <filename>** : This command increases the brightness of the entire image.

II. **value-component <fileName> <newImageName**>: This command converts the image to greyscale using the value component.

III. **red-component<fileName> <newImageName>:** This command converts the image to greyscale using the red component.

IV. **green-component<fileName> <newImageName>**: This command converts the image to greyscale using the green component.

V. **blue-component<fileName> <newImageName>:** This command converts the image to greyscale using the blue component.

VI. **intensity-component<fileName> <newImageName>**: This command converts the image to greyscale using the intensity component.

VII. **luma-component<fileName> <newImageName>:** This command converts the image to greyscale using the luma component.

VIII. **sharpen <fileName> <newImageName>:** This command sharpens the image.

IX. **sepia <fileName> <newImageName>**: This command sepia filters the image.

X. **greyscale <fileName> <newImageName>:** This command greyscale filters the image.

XI. **blur <fileName> <newImageName>:** This command blurs an image.

XII. **load <filePath > <filename>:** This command loads an image from a file in PPM format. Replace <filename> with the name of the file to load.

XIII. **save <destinationPath> <filename>:** This command loads an image from a file in PPM format. Replace <filename> with the name of the file to load.

-file <script filePath>: To run a script text file.

This allows you to load up a text file that has commands line-by-line. It will execute the commands and exit the program. To make sure it worked, look at where you wanted the altered image to be saved at.

<space/pressing enter without input>: To use GUI mode.

Refer to the USEME file that goes through how to use this mode.

Which Part of the Program Are Complete:

All the modes are complete. You can successfully use the following modes listed above.

Design Changes/Justification:

There weren't any design changes.

To create GUIView and GUIController, I included the addition of separate interfaces for the two. The interfaces are IGUIView and IGUIController. These two interfaces include the methods used by the corresponding classes.

These two classes are then called in the main when the command to implement the GUI mode. In regard to the previous code for the script and the text mode, there was nothing made.

The change that was caused was adding to the ImageUtil class. This class was to include methods that can be used for utility methods for the Images used in the Assignment. The method added was:

Public static BufferedImage convertToBufferedImage(IImageState image)

This method is to convert the given image to a buffered image. This is used in the GUIController to allow the user interaction to be clearer with painting the picture.

This method, however, isn't one that changes design, it just adds onto the design of separating these utility methods.

Citations:

The image used in my USEME, and is included in my res for testing (Godzilla.jpg).

https://www.peakpx.com/en/hd-wallpaper-desktop-aktix