USEME

Command Line Level Input

- If one wants to have a text-based script as input, they should put "-file" for the first argument, indicating they will pass a script to the program. The file path should be the second argument.
- If one wants to interact with the program by typing command during the session, they should put "-text" for the first argument in the command line, indicating they will interactively manage and process the image through inputting text.
- If one wants to interact with the program through GUI, run the program without any arguments.

†Error Handling

- Insufficient command line level arguments will cause a system level exception with alerting message, user have to run the program again with correct inputs.

Text-mode : Program Level Input

- Supported Commands
  - Currently, this program supports the following commands, and user must follow the syntax to use them.
    - load [source] [name]
    - save [destination] [name]
    - brighten [value] [name] [new name for modified]
    - darken [value] [name] [new name for modified]
    - [red/green/blue/value/intensity/luma]-component [name] [new name for modified]
    - [horizontal/vertical]-flip [name] [new name for modified]
    - blur [name] [new name for modified]
    - sharpen [name] [new name for modified]
    - greyscale [name] [new name for modified]
    - sepia [name] [new name for modified]

        - size
    - An image can be processed, obviously, only if it's been loaded to the library using `load` command, with its identifiable name.

- Commands above can be typed in line by line, with comment lines start with `#` be ignored. User can also give multiple command on the same line, using splitter `&` to separate. In this case, the command will be executed linearly, and if one fails, the error message will be thrown immediately and no longer process the remaining commands, asking users to correct the syntax and try again.
    - Restrictions: Inspired by the terminal.app on macOS, we do the same thing to ask user not to put splitter at the beginning of the line as it does not make sense to do nothing and then do something else, yet we allowed the splitter is put at the end of a line as users may want to execute a series of command but put the remaining of commands (after the splitter) on a different line.
- Notes: multiple spaces between arguments are allowed and will be ignored, number of arguments need to be exact, both extra/insufficient input will cause a complaint from the program.
- Quit Command
    - To minimize the chance that user accidentally quit the program or throw away the unperformed operations, we ask user to use the capitalized `QUIT` to quit the program, and this operation has to be the **last** argument on a line.

† Error Handling

- All the following will cause a complaint from the program. The program will print specific informative message without quitting itself and allows users to try again.
    - Insufficient arguments for a command.
    - Too many arguments for a command.
    - User types in an unknown command.
    - User doesn't specify the extension of the image file when loading.
    - User gives an invalid file name which starts with "." and immediately followed by a file extension when loading.
    - User load an image with unsupported extension.
    - User provide an image that cannot be found from the machine's file system.
    - User try to process a non-existing file in the library.
    - When darkening/brightened an image, users fail to specify the value of the adjustment.
    - Try to save a file to a read-only directory.
    - The process of loading a file is interrupted.
    - The process of saving a file is interrupted.

Script-mode: How to run the script

The command script and example images are saved (and zipped) in the `res/imagesAndScript/` directory. The text script is named as `ImageProcScript.txt`. Graders can run this file under the res/imagesAndScript/ by inputting in the command line

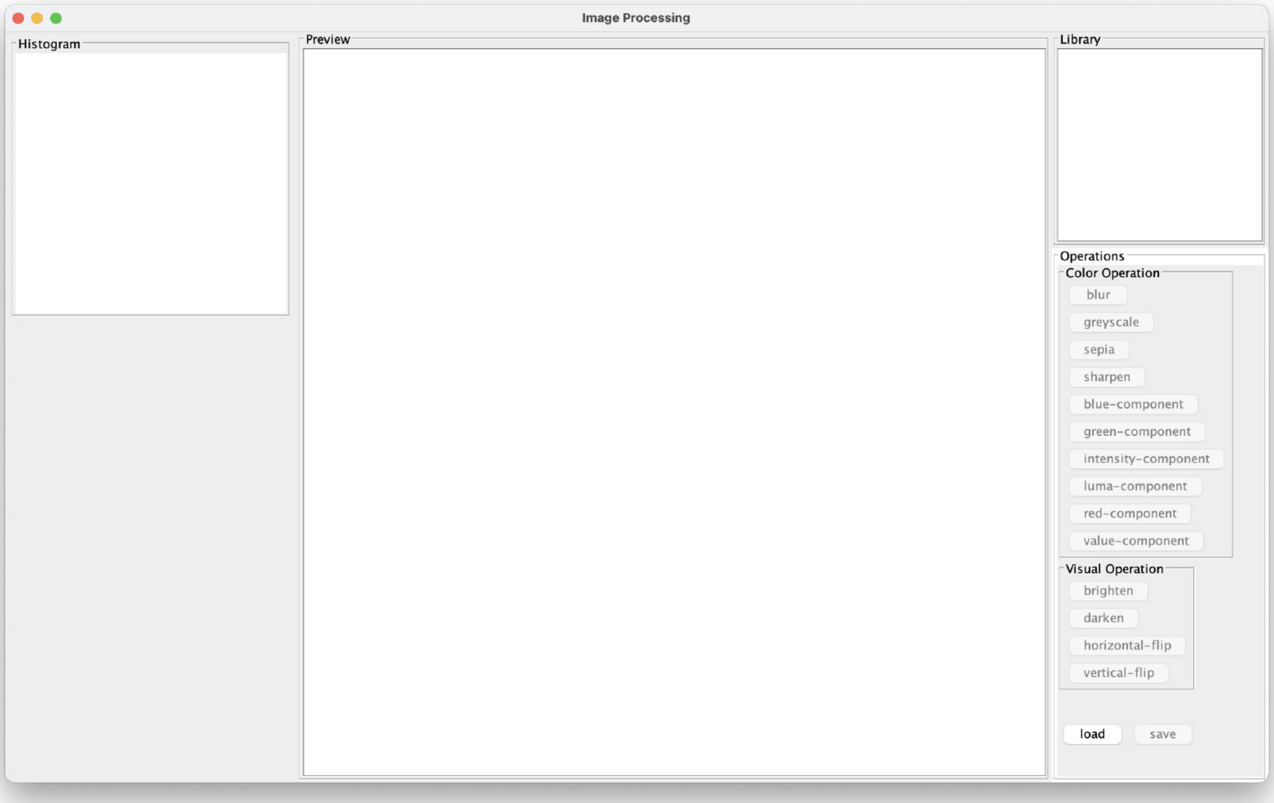
```
imagesAndScript % java -jar ../HW06.jar -file ImageProcScript.txt
```

† Error Handling

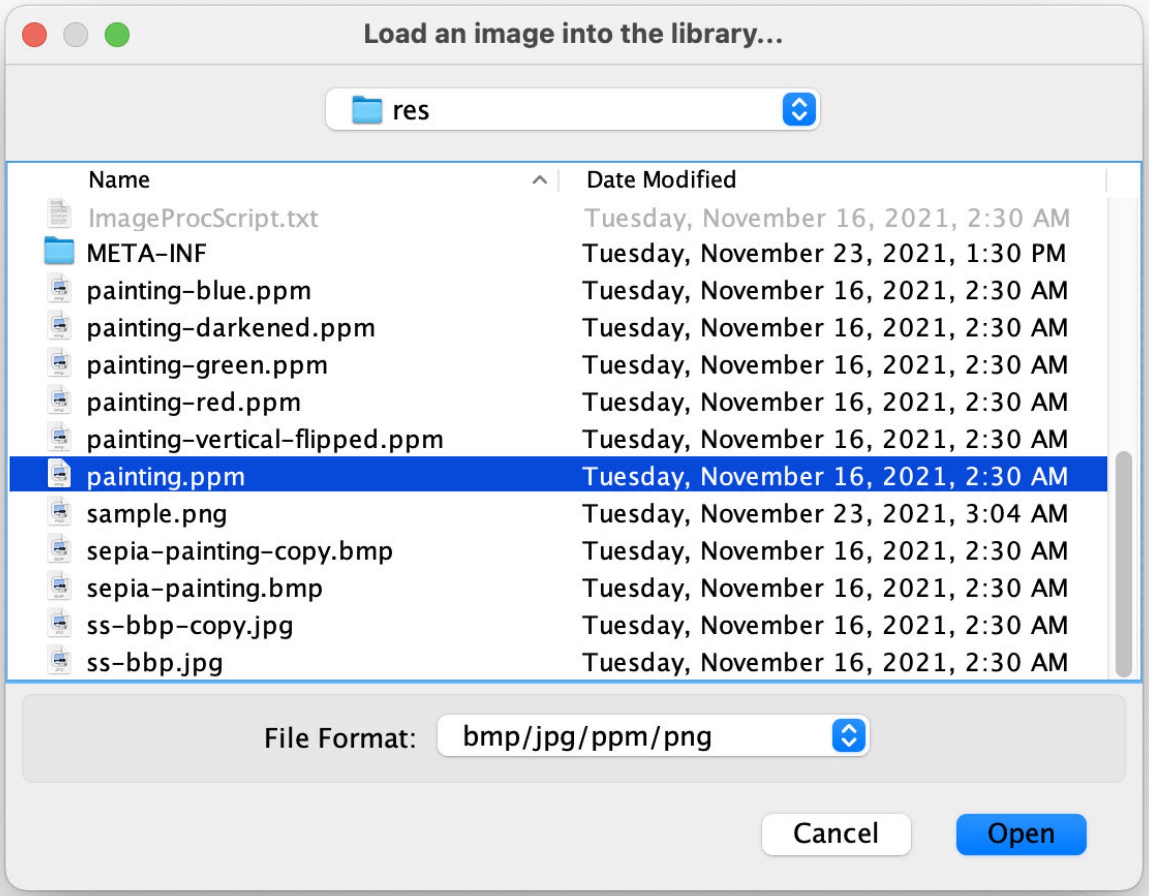
- Same as §Text-mode, see above.

GUI-mode: Tutorial

1. StartPage:



2. Press "load" to import an image:



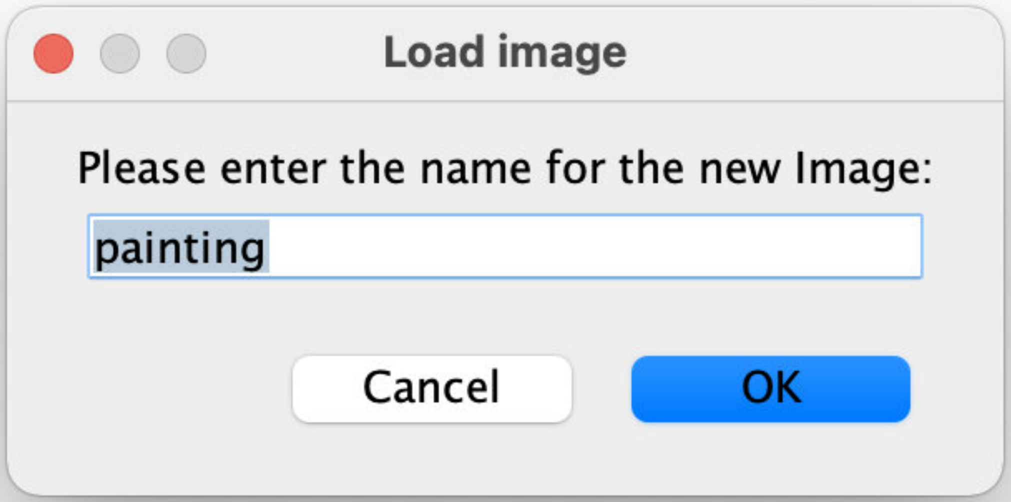
Load an image into the library...

res

Name	Date Modified
ImageProcScript.txt	Tuesday, November 16, 2021, 2:30 AM
META-INF	Tuesday, November 23, 2021, 1:30 PM
painting-blue.ppm	Tuesday, November 16, 2021, 2:30 AM
painting-darkened.ppm	Tuesday, November 16, 2021, 2:30 AM
painting-green.ppm	Tuesday, November 16, 2021, 2:30 AM
painting-red.ppm	Tuesday, November 16, 2021, 2:30 AM
painting-vertical-flipped.ppm	Tuesday, November 16, 2021, 2:30 AM
painting.ppm	Tuesday, November 16, 2021, 2:30 AM
sample.png	Tuesday, November 23, 2021, 3:04 AM
sepia-painting-copy.bmp	Tuesday, November 16, 2021, 2:30 AM
sepia-painting.bmp	Tuesday, November 16, 2021, 2:30 AM
ss-bbp-copy.jpg	Tuesday, November 16, 2021, 2:30 AM
ss-bbp.jpg	Tuesday, November 16, 2021, 2:30 AM

File Format: bmp/jpg/ppm/png

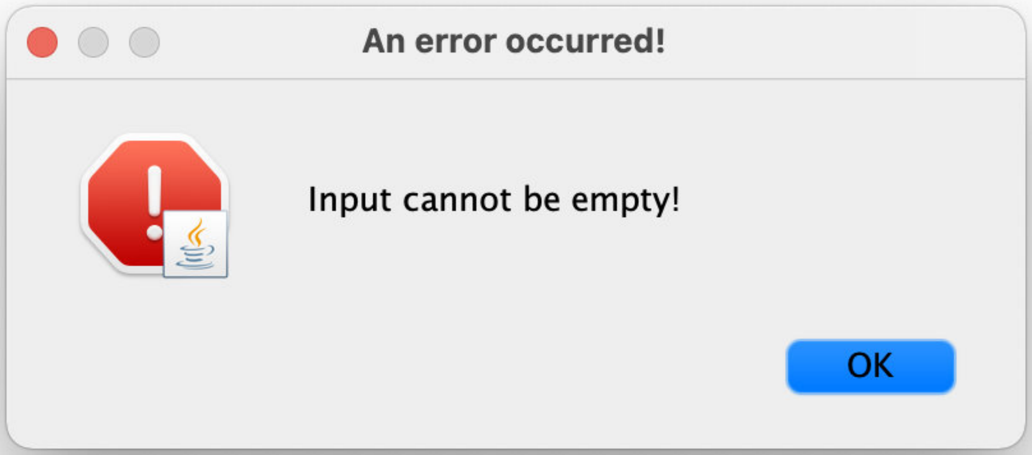
3. Import painting.ppm from res/:



Please enter the name for the new Image:

painting

- Empty input will lead to an alert window saying "Input cannot be empty"



Input cannot be empty!

4. Use the default name for the newly imported image "painting":

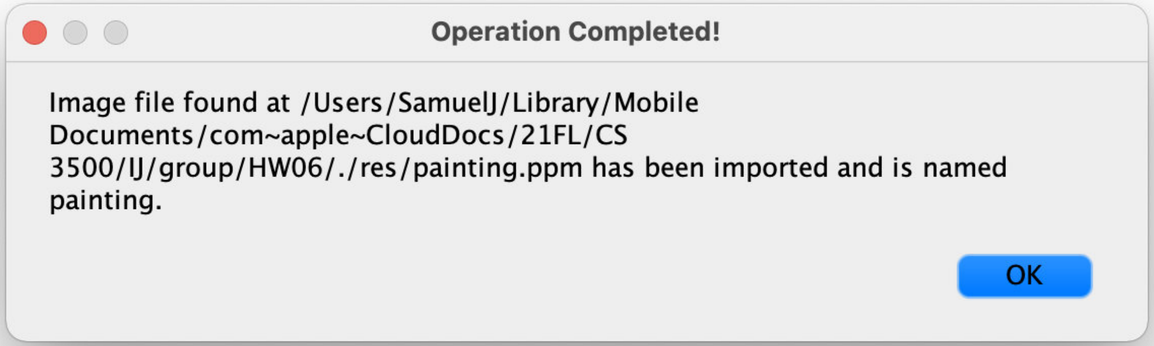
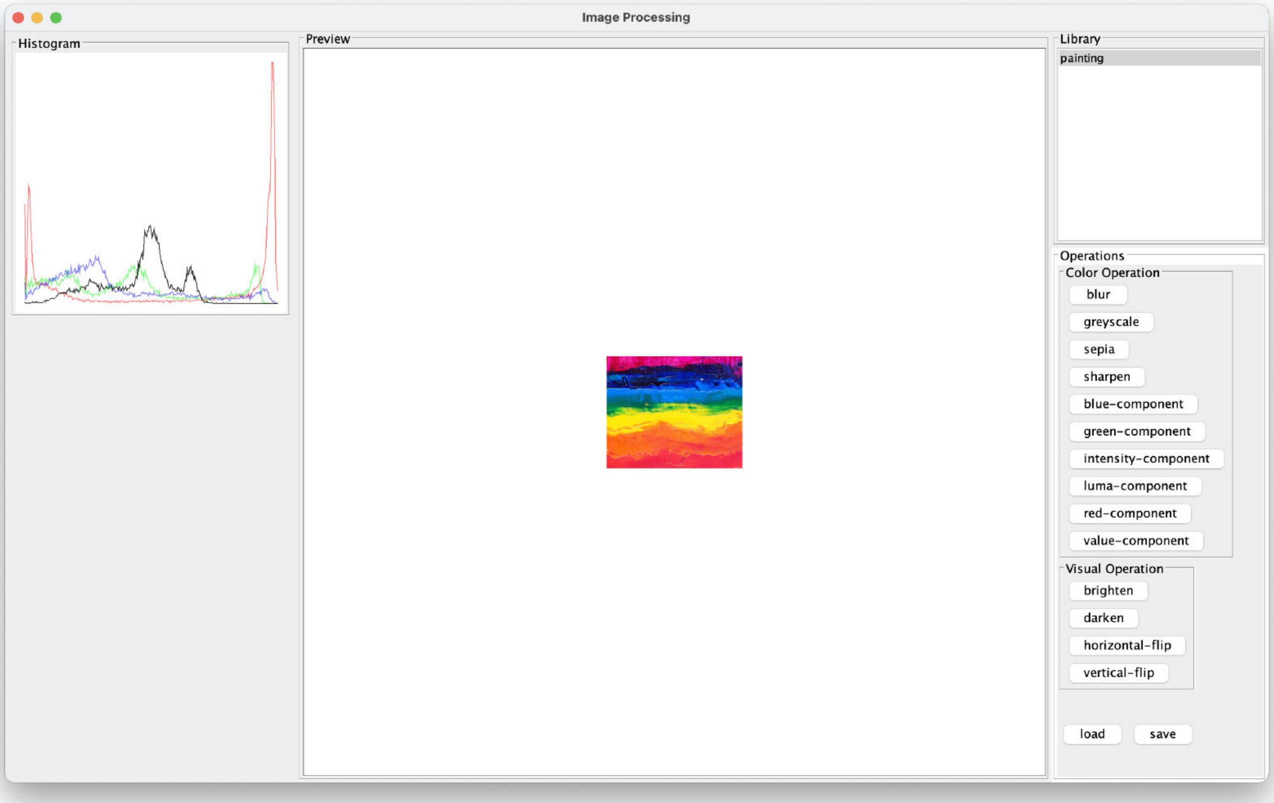

Operation Completed!

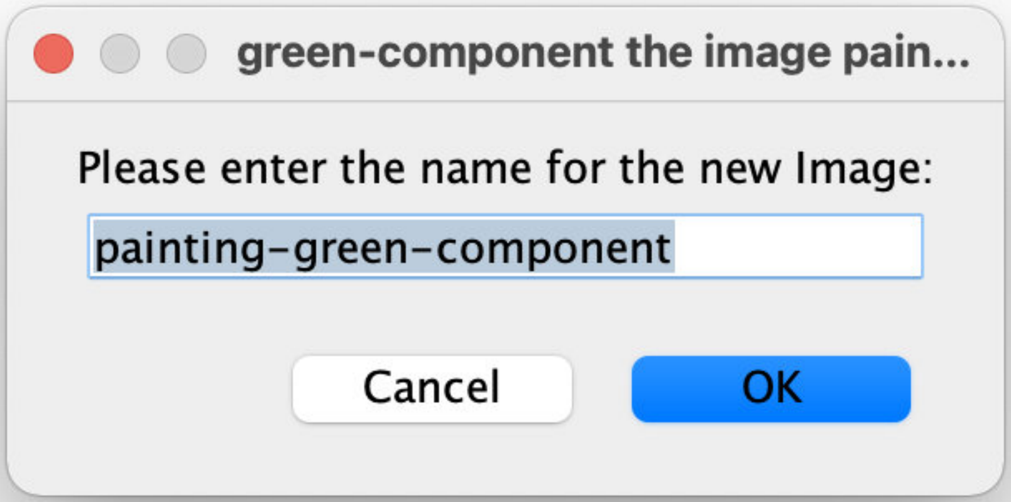
Image file found at /Users/SamuelJ/Library/Mobile Documents/com~apple~CloudDocs/21FL/CS 3500/IJ/group/HW06/./res/painting.ppm has been imported and is named painting.

5. Image Preview & Histogram:



- program will always auto select to the newly imported image after a load operation

6. Press green-component button, rename the newly created image:

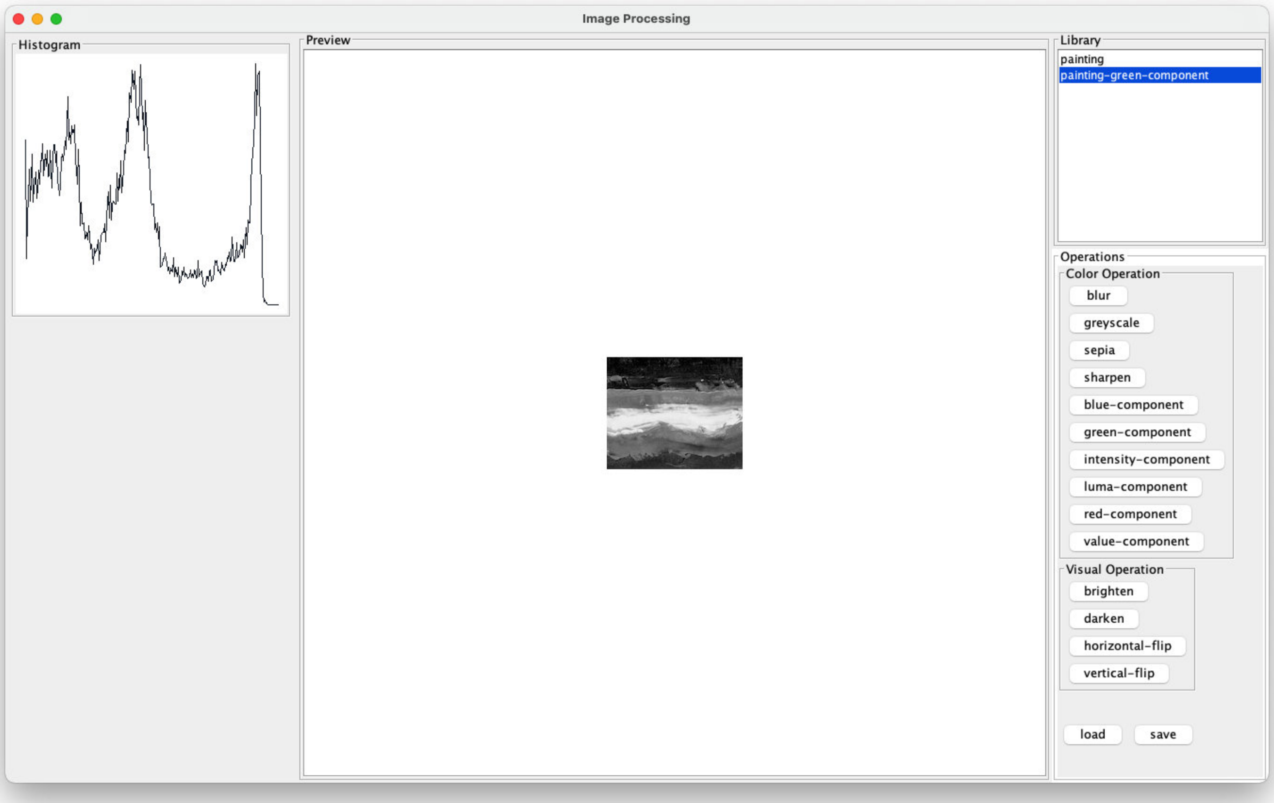

green-component the image pain...

Please enter the name for the new Image:

painting-green-component

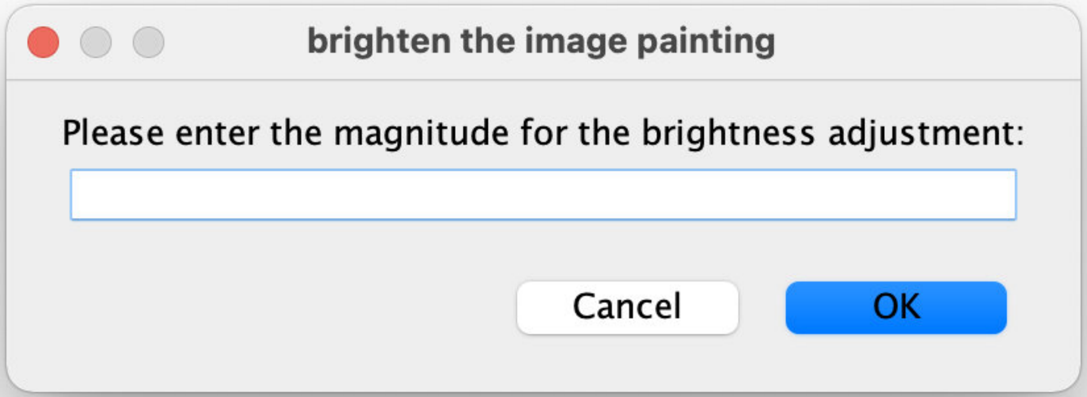
- Program always append the current operation to the name of the current processing image and use it as the default name of the newly created image.

7. Green-component image preview:



- program will always, again, auto select to the newly created image after an operation.

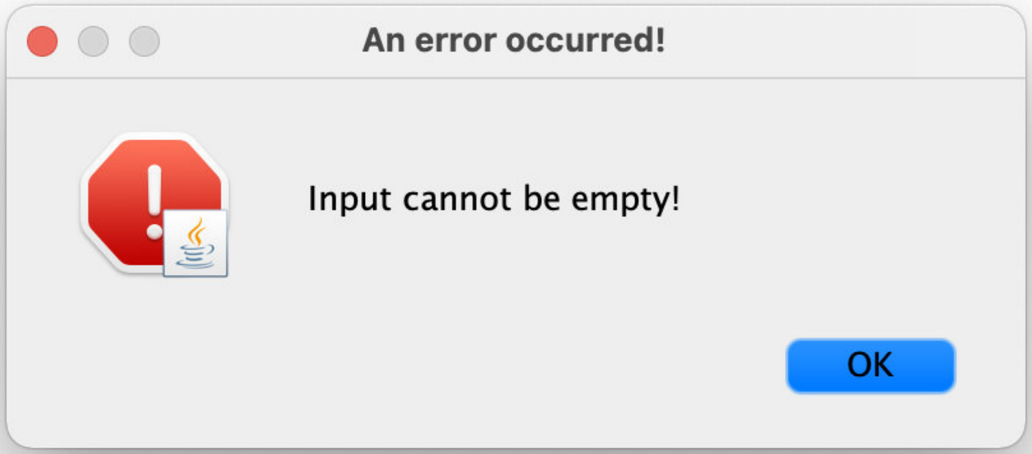
8. brighten image "painting":



brighten the image painting

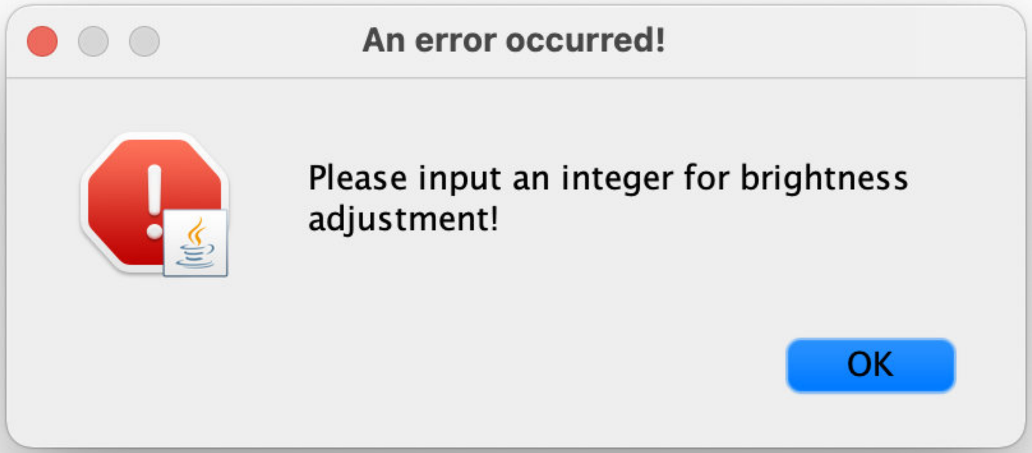
Please enter the magnitude for the brightness adjustment:

- No default value for magnitude as it highly depends on user's decision
- If the input is empty, alert window will pop up:



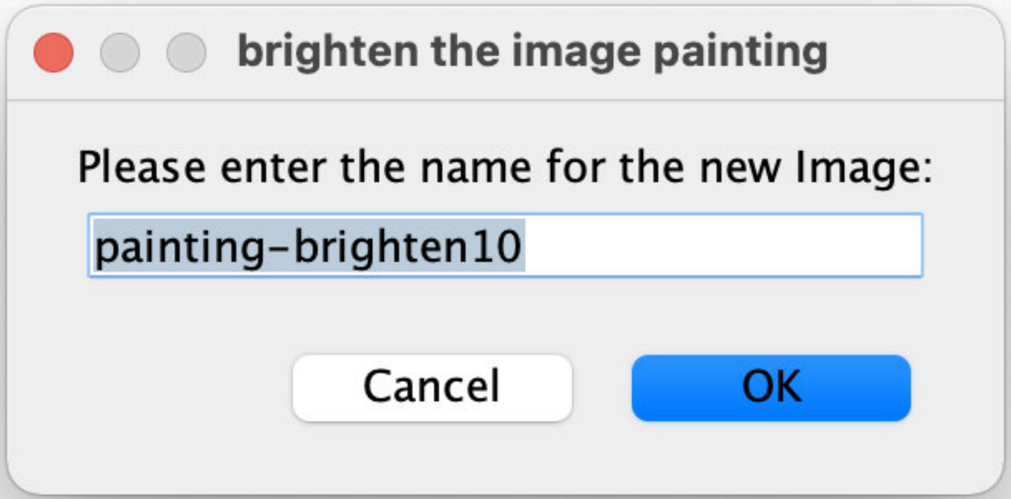
Input cannot be empty!

-
- If the input is not pure number, another alert window will pop up:



Please input an integer for brightness adjustment!

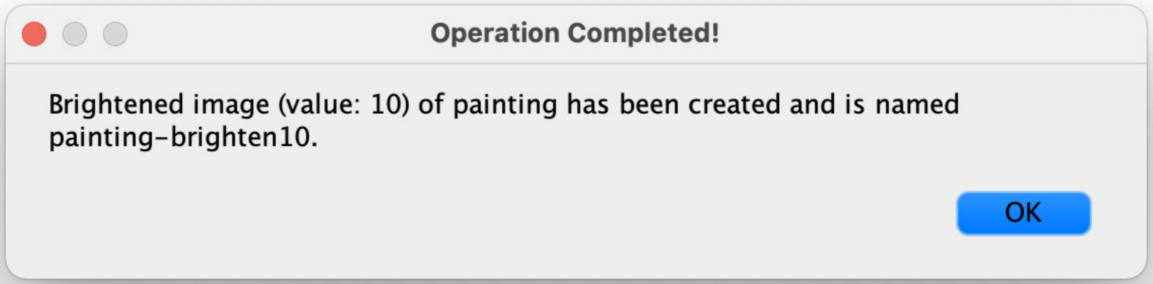
9. Rename the brightened image, auto-naming includes the adjustment magnitude:


brighten the image painting

Please enter the name for the new Image:

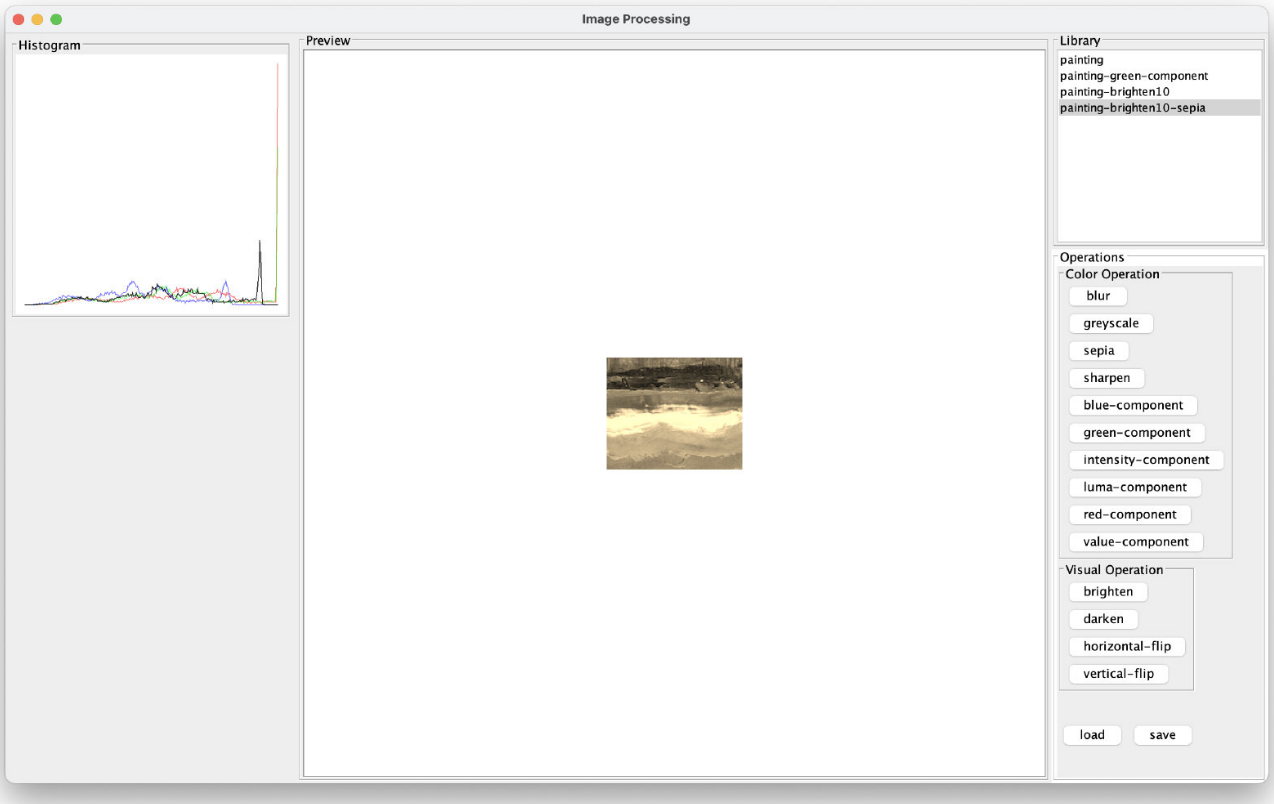
painting-brighten10

10. Feedback after brighten:


Operation Completed!

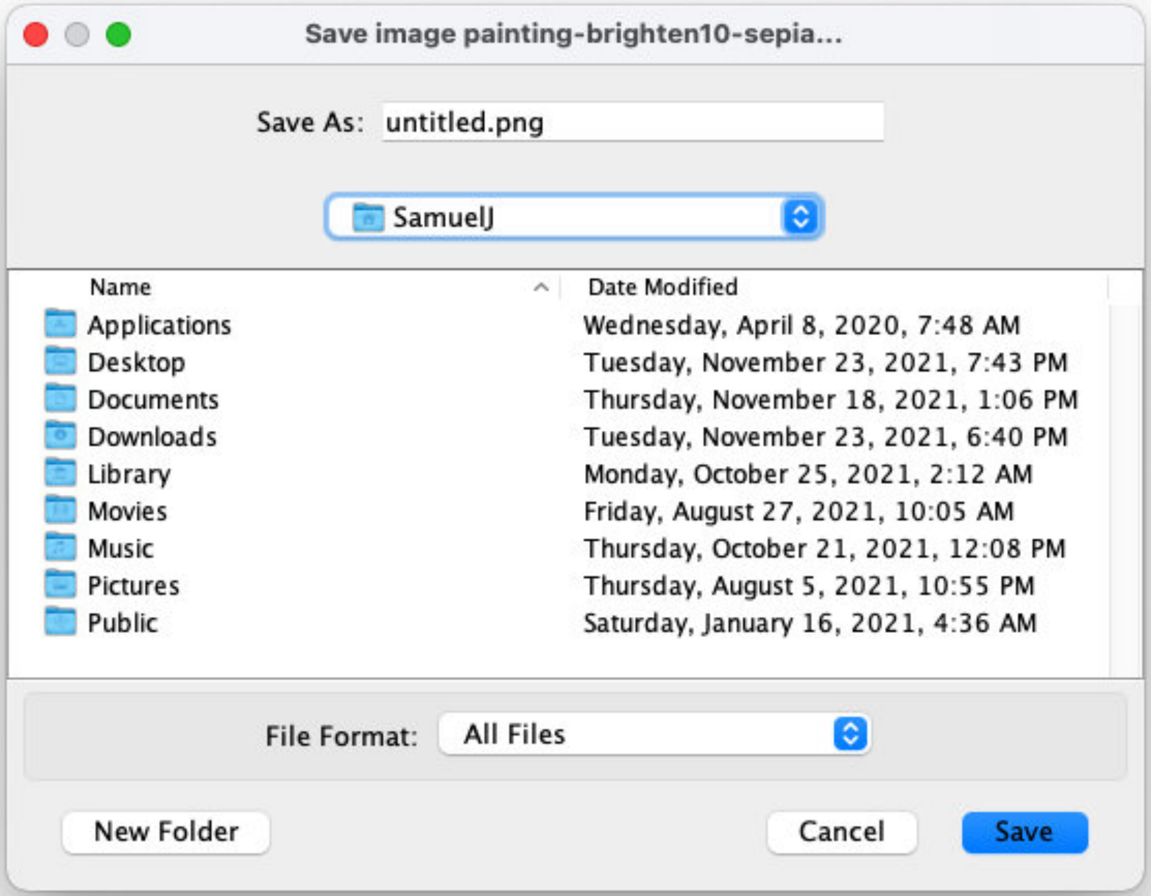
Brightened image (value: 10) of painting has been created and is named painting-brighten10.

11. After doing sepia on the latest image, we have the view window as follows:



painting
painting-green-component
painting-brighten10
painting-brighten10-sepia

12. Try to save this image (by pressing save button):



Save image painting-brighten10-sepia...

Save As: untitled.png

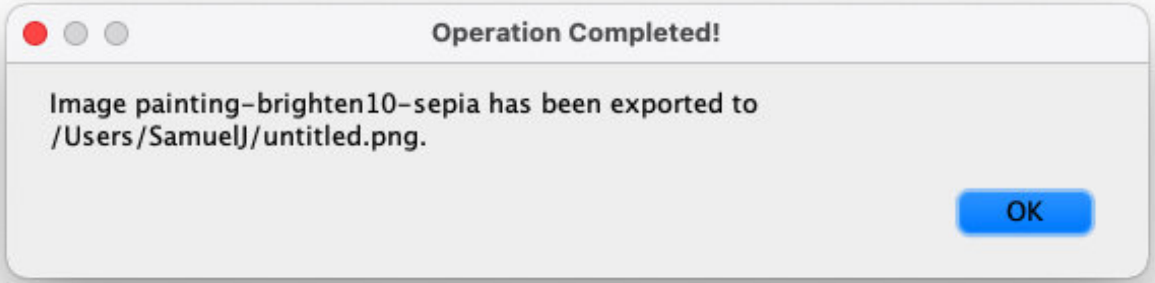
SamuelJ

Name	Date Modified
Applications	Wednesday, April 8, 2020, 7:48 AM
Desktop	Tuesday, November 23, 2021, 7:43 PM
Documents	Thursday, November 18, 2021, 1:06 PM
Downloads	Tuesday, November 23, 2021, 6:40 PM
Library	Monday, October 25, 2021, 2:12 AM
Movies	Friday, August 27, 2021, 10:05 AM
Music	Thursday, October 21, 2021, 12:08 PM
Pictures	Thursday, August 5, 2021, 10:55 PM
Public	Saturday, January 16, 2021, 4:36 AM

File Format: All Files

- Image will always have a default name "untitled.png" as the file name, yet user can edit the file name however they want, as well as the file extension as long as it's supported by the program (currently, we support png/jpg/ppm/bmp)

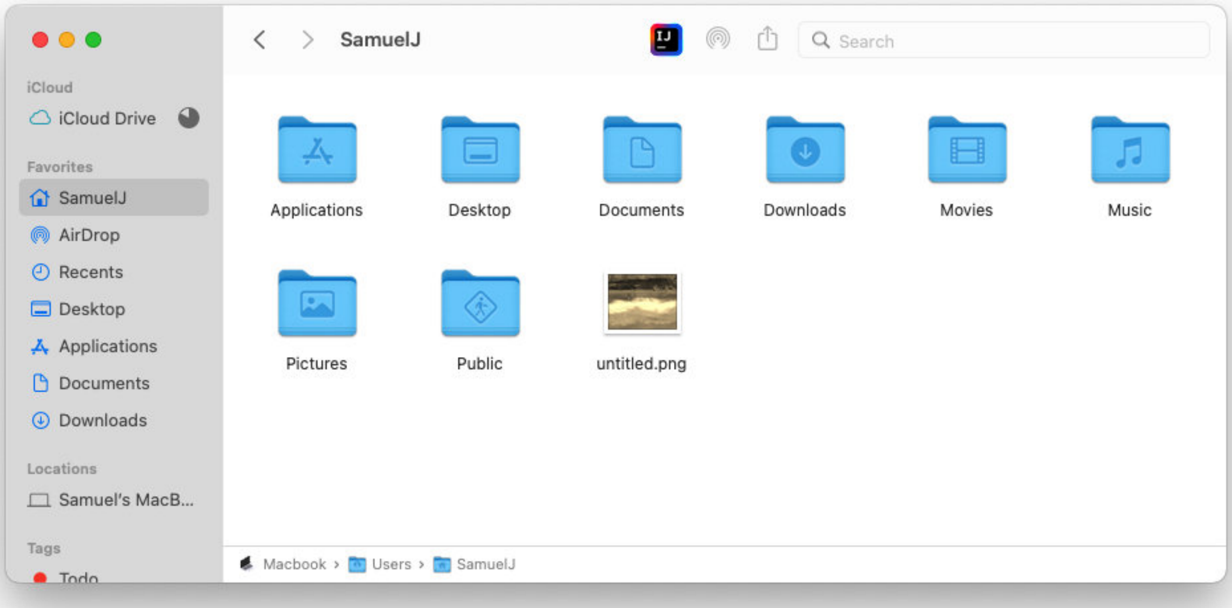
13. Save succeed feedback:



Operation Completed!

Image painting-brighten10-sepia has been exported to /Users/SamuelJ/untitled.png.

14. Inspect saved image:



† Error Handling

- The tutorial example above has already presented a few.
- Basically, errors will be thrown under the same circumstances as the §Text-mode mentioned, yet GUI supports to render an error with a pop-up alert window.