

Programming Exercise: Batch Grayscale and Image Inversion

Assignment 1: Batch Grayscale

You have learned how to convert an image to grayscale, and how to select and process several images to convert them to grayscale and display them. You also learned how to copy an image and save it with a different filename. Now put this all together in one program that batch processes several images, and creates and saves new images (with new filenames) that are grayscale versions of each image.

More specifically,

- Your program should let the user select multiple image files
- For each image,
 - Create a new image that is a grayscale version of the original image
 - Save the grayscale image in a new file with the same filename as the original image, but with the word “gray-” in front of the filename. For example, if the original file was named `lion.png`, the new image would be a grayscale image and be named `gray-lion.png`.

Hint: Start with the Batch Grayscale program that processes many images, and add in code to save those files with new names.

Assignment 2: Image Inversion

Write a program to create new images that are photographic negatives (or inverted images) of selected images and save these new images with filenames that are related to the original images, such as adding “inverted-” in front of the old filename. In inverting an image, a pixel’s red, blue, and green components are modified to be the exact opposite within the 0 to 255 range. That is, if a pixel’s red, blue, and green values are (34, 198, 240), then that same pixel in the inverted image would have the red, blue and green values of (221, 57, 15). Note that $255 - 34$ is 221, $255 - 198$ is 57, and $255 - 240$ is 15.

For example, the image on the right is the inverted image of the image on the left.



Here is another example. The second image below is the inverted image of the first image.



To tackle this problem

- First think about how to create a new image that is the inverse of another image.
 - Start by solving an example by hand, write down the steps, find patterns and then test the algorithm
 - Then create a new BlueJ class called `BatchInversions`. In this class create a method named `makeInversion` that has one parameter, an image, and returns a new image that is the inverse of the original image.
 - Then add another method called `selectAndConvert` to handle the batch processing of files. This method allows the user to select several files and display the images. Test this method to make sure it works.
 - Then modify the `selectAndConvert` method to save each inverted image as a file with a new filename. Use the old filename and append to the front of it "inverted-"