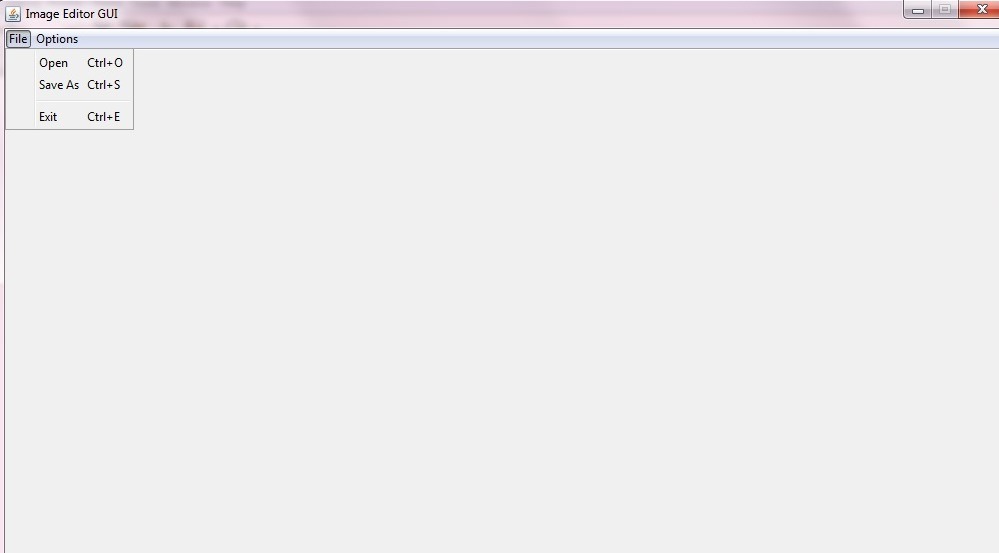
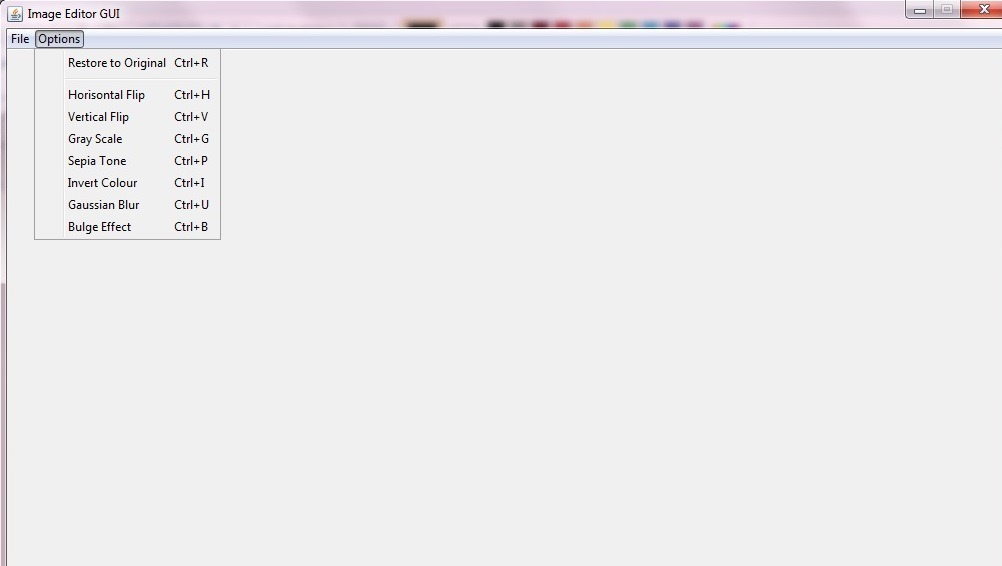
Adobe Photoshop is a commonly used application for image manipulation. While many of its image effects are complicated, many can be programmed with relative ease. Your job is to create a GUI in Java that loads images and allows users to perform the following image manipulations:

1. Horizontal flip
2. Vertical flip
3. Gray scale
4. Sepia tone
5. Invert color
6. Gaussian blur
7. Bulge effect

**Part A: Design**

Your interface **must** look like the following images:





You may assume that images will be no larger 1000x1000 pixels. You will find [this](http://docs.oracle.com/javase/tutorial/uiswing/components/index.html) GUI tutorial beneficial.

**Part B: File Menu**

This menu will provide users with the following three options:

1. Open: This option will allow users to open any image located on their computer’s file system.
2. Save As: This option will provide users the ability to save their images anywhere on their computer’s file system.
3. Exit: This option will shut down the program.

**Part C: Options Menu**

This menu will provide users with the following eight options:

1. Restore to Original: Reverts the current image to its original state.
2. Horizontal Flip: Reflects the image about the y-axis.
3. Vertical Flip: Reflects the image about the x-axis.
4. Gray Scale: Converts the image to black-and-white. See [here](http://en.wikipedia.org/wiki/Grayscale).
5. Sepia Tone: Gives an image an old photo look; Google it!
6. Invert Colour: Inverts the colors in an image; Google it!
7. Gaussian Blur: Refer to [this](http://pixelstech.net/article/1353768112_Gaussian_Blur_Algorithm) guide; please look past the broken English.

<http://www.ruanyifeng.com/blog/2012/11/gaussian_blur.html>

<https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf>

1. Bulge Effect: Causes the image to bulge from the center. This is done by converting every pixel location, in Cartesian form, to polar coordinates. Then the following transformation is applied:



A k-value greater than 1 causes the bulge effect, but it must be compensated by m for larger values of k.

<https://www.mathsisfun.com/polar-cartesian-coordinates.html>

<https://math.stackexchange.com/questions/266250/explanation-of-this-image-warping-bulge-filter-algorithm>

**Tips:**

You will find the classes [ImageIO](http://docs.oracle.com/javase/1.4.2/docs/api/javax/imageio/ImageIO.html) and [BufferedImage](http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/image/BufferedImage.html) invaluable for this project. Also, ensure that your BufferedImages are using the TYPE\_INT\_RGB mode. This mode allocates certain bits in an int for red, green and blue. You can manually enter your desired RGB values using the Color class as well. See the setRGB and getRGB methods for more details.

**Due Date:** Friday, June 8, 2018