

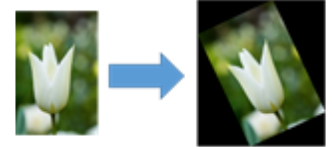
In this assignment, your task is to create a GUI program as an image collage maker. To start with, collect a few images that you'd like to put in your collage. An example collage is shown to the right. Well, yours might not be this fancy, but you can always think about how to add extra features.



You will need the following in your program:

- Allocate a blank RGB image in your program to hold the collage. You can use a preset size for now.
- An **axes** to show your collage. This allows the editing of the collage in a WYSIWYG manner. Note: You can place your **axes** inside a **panel**, as in the lab task, to allow a larger collage.
- A **list box** containing the filenames of the images to put in the collage.
 - ◆ The filenames are added one by one. To do this, you need to provide an **Add pushbutton**. The user gets to select a file when clicking this **pushbutton**. Note: In your file selection **dialog box**, you should limit the file type to valid image types. Once an image file is added, also load the image itself into your program. (Hint: You can choose one source image as the background or the collage.)
 - ◆ Also provide a **Delete pushbutton** so that you can remove an image in the list.
 - ◆ When drawing the images onto the collage, follow the order in this **list box**.
 - ◆ Provide some visual feedback to indicate the currently selected image, such as a colored bounding box around the image. This should be updated every time the selection is changed.
 - ◆ Add **Up** and **Down pushbuttons**. Clicking them moves the selected filename up or down in the list. This allows the user to change the order of drawing the images.
- The "editing features" that control how a source image is drawn in the collage.
 - ◆ The required ones: Position (top-left corner) and size (width and height). Note: The size can be different from the size of the source image.
 - ◆ Optional features: Rotation (in degrees), frame (width in #pixels of an extra boundary around a source image), cropping (you need to specify a cropping rectangle for the source image), etc. You can add other editing features if you want to.
 - ◆ In the program, it's best to collect all the editing features of an image, together with the image itself, in a structure.
 - ◆ When a new source image is added, your program should assign default values of the editing features.
- A few **edit boxes** for modifying the editing features.
 - ◆ Whenever a source image in the image **list box** is selected, the corresponding editing features should be displayed in these **edit boxes**.
 - ◆ (Optional) You can try to replace the **edit boxes** with **spinners** to allow fast incremental changes to the editing features with small up/down buttons.
- How to draw the source images onto the collage:
 - ◆ Create an edited version of each source image first. You can use toolbox functions such as **imresize**, **imrotate**, **imcrop**, etc. for this task. Do not directly modify the source images.
 - ◆ When copying the edited images onto the collage, be careful about parts of the image that might fall outside of the collage. You need to crop those parts out before copying.
 - ◆ Special care when rotating source images: The function **imrotate** generates images that contain background pixels (pixels that are not part of the original image), shown as black regions in the

illustration to the right. To avoid copying the background pixels to the collage, what you can do is: Create a binary image the same size as the source image and set all its pixels to `true`, then apply `imrotate` to this image as well. When copying the rotated source image, use this rotated binary image as a logical mask, copying only those pixels that are `true` in the mask.



- A `pushbutton Update` for redrawing the collage after some modifications.
 - ◆ To allow for more instant updates after you modify the editing feature values via the `edit boxes`, implement a separate function `update_output` to be called by the `ValueChanged` callbacks of the `edit boxes` or `spinners`. Let the callback of the `Update pushbutton` call `update_output` as well.
- A `pushbutton SaveCollage` for saving the collage as an image file. This should bring up a `dialog box` for path and filename specification.

Submission:

Submit your code through E3. Name your file `P5_#####.mlapp`, where the `#####` represents your student ID. There will be a three-day grace period after the due date, during which there will be a 10%/day deduction for your grade.

For a GUI program, an easy-to-use layout is important. So 5% of your grade will be based on your GUI design, although this part is to be judged subjectively by the TAs.

In addition to the program, also submit two image files: (1) A collage created using your program. (2) A screenshot of your GUI taken while editing the submitted collage. Name these two images `P5_#####_out.jpg` and `P5_#####_gui.jpg`, respectively. These image files will not be graded for their qualities; instead, they are intended to prove that your program is actually usable. There will be a 10% deduction of your grade for failing to submit these images.

Your code should include sufficient comments. This will be part of the grade. Include your name and ID at the top of your code.