MATLAB Programming NCTU Fall 2017 Assignment#3 due 2017/12/7

In this assignment, you will implement a picture viewer program using the simple approach of user interaction. The functionalities of this program will be similar to those of the picture viewer that comes with Windows.

- The only input to the main function is a path to a folder (not a file). You need to call **`dir`** to retrieve the list of picture files in that directory. (For simplicity, limit the file extension of "**`*.jpg`**" for now.) Note: **`dir`** returns a structure array; check the documentation on how to get the actual list of file names.
- When the program starts, display the first picture in the list. Let the user move between picture using the '**<**' and '**>**' keys. If the folder contains no picture file, just display a black rectangle.
- Use the function title to show the information of the current picture. An example regarding the format: **`Picture 3 of 8 (picture_file_name.jpg) 320x240 @100%`**
- Use a fixed-size array (such as 640x480x3) as the input to **`imshow`** so that the figure does not change size for each new picture. When an image is first displayed, if it is smaller than this size, pad with zeros so that the actual image appears at the center, and if it does not fit into this size, resize it so that it fits within the display region. Indicate the scaling factor in the title. (Note: A convenient function to use here is **`imresize`**, which we will explain in the class.)
- The next functionality allows the user to see the local detail around a point in the image. When a point within the image is clicked, bring up another figure that shows an enlarged version of a square region centered at the clicked point. You can determine for yourself the size and scaling factor of this "highlight" region.
  - You have to handle the case when the square is partially outside of the image.
  - After you display the extra figure, call **`figure`** to move the focus back to your original figure.
  - Explicitly set the positions of the two figures so that they do not block each other.
  - Draw a square on the displayed image to indicate the region shown in the highlight figure.
- Allow the user to change the size of the highlight square with the '**+**' and '**-**' keys. You can set lower and upper bounds of the size. The scaling factor used to generate the image shown in the highlight figure needs to be changed accordingly.
- Allow the user to use the arrow keys to move the highlight square to different places in the image. Note: The center cannot be moved to outside of the image.
- If the user clicks in the figure anywhere outside of the image, or when a different image is selected (by using the '**<**' and '**>**' keys), close the highlight figure and remove the highlight square from the image.

Submission: Submit your code (m file) through e3. Name your file **`P3_#######.m`**, 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.

A "copy detection" will be applied to your submissions, and those found to have copied assignments will receive zero points for the assignment.

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

There will be demo session with the TAs (date/time to be announced later).