MATLAB Programming NCTU Spring 2017 LAB#9 2017/5/5

This lab lets you try to interaction with a figure window without using GUI objects. You will write a program that has some drawing capability. Specifically, your program should do the following:

- Create a figure and place a blank axes in the figure. You can just use the **`figure`** and **`axes`** functions to do this. The initial axes contains no plot. Set its axis limits to 0 to 1 along both directions using **`axis`**. Also use **`axis equal`** so that the circles you will draw actually appear circular.
- Go into an infinite loop (use **`while 1`**) in the program. Within the loop, wait for a user event before proceeding. Do the following depending on the event:
  - For a mouse-click event, draw a circle at the location of the mouse click if it is within the axes limits. The radius, line width, and color of the circle are set via user inputs. (You can specify the initial radius, line width, and color before going into the loop.)
    - You can use **`get(gca,'CurrentPoint')`** to get the coordinates of the clicked point in the axes coordinate system, so you do not need to do the coordinate transform.
    - To draw a circle, you can compute the point coordinate on a circle (using something like **`r*cosd(0:360)`** and **`r*sind(0:360)`**) and use **`plot`** on these points to draw them. The line width and color properties are used within the call of **`plot`**.
    - You can use **`hold on`** to retain all the circles you have drawn.
  - For a key-press event, do one of the following based on the key pressed:
    - Digits '**`1`**' - '**`9`**': Set the line width to the number.
    - One of the characters that represent colors, '**`r`**', '**`b`**', '**`k`**', '**`w`**', '**`y`**', '**`c`**', '**`m`**', '**`g`**': Use the selected color for subsequent circles.
    - '**`+`**' and '**`-`**': Increase and decrease the circle radius by 10%, respectively.
    - Capital '**`X`**': Clear the drawing. You can do this using the **`cla`** statement.
    - '**`q`**': Exit the loop. You can use the **`break`** statement here.
    - Do nothing for any other keys, unless you define some additional behaviors (such as changing shapes, etc.) yourself.
- To let the user keep track of the current parameters, display a text string that lists the parameters using the function **`text`**.