

Multimedia systems

Lecture 6

LECTURER

MOUHSEN IBRAHIM

Contents

- ▶ Introduction to creating UI in Matlab
- ▶ Using GUIDE to create a UI
- ▶ Your first UI.
- ▶ Place the components
- ▶ Call-Backs
- ▶ Another Example
- ▶ Screenshots
- ▶ More information

Introduction to creating UI in Matlab

- ▶ A User Interface is one or more windows containing controls called components that enable users to perform interactive tasks.
- ▶ UI components include menus, toolbars, push buttons, radio buttons, list boxes, sliders and more...
- ▶ A UI typically waits for actions to be performed by users then interact to them using call-backs, this kind of programming is called event-driven programming, call-back execution is asynchronous.
- ▶ In Matlab a UI consist of two files a .fig file which is a binary file that contains component descriptions and placement.
- ▶ A code file which contains Matlab code for call-backs.

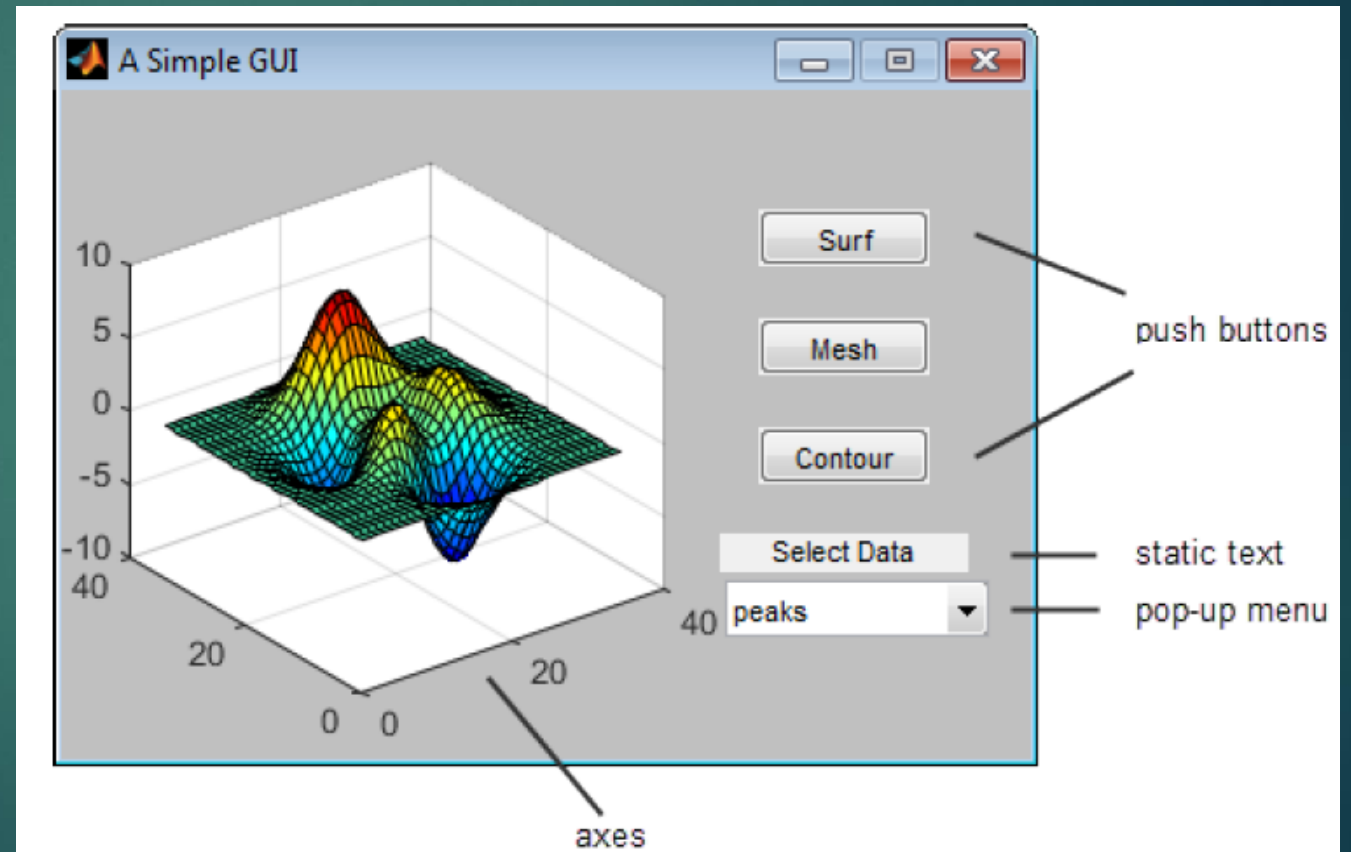
Using GUIDE to create a UI

- ▶ GUIDE can be used to create UIs in Matlab, it creates two files for each UI and enables the user to use a graphical interface to define and place components.
- ▶ To start GUIDE execute the following command in Matlab: `guide`
- ▶ Create a new Blank GUI.
- ▶ Now the main window for GUIDE appears.
- ▶ To show components names in the component palette to the left select File -> Preferences -> GUIDE
- ▶ Select Show names in components palette and click OK.
- ▶ Now we can start creating a new UI.

Your first UI

The image to the right shows an example UI that we will create.

This UI contains push buttons, static text, pop-up menu and an axes component.



Place the components

- ▶ Use the command palette to place three push buttons, a static text, pop-up menu and an axes.
- ▶ Left click a push button and select Property Inspector.
- ▶ Change the String property to Surf, Mesh, Contour for all push buttons.
- ▶ Change the static text String property to select data
- ▶ Change the pop-up menu String property to peaks, membrane and sync, each on a separate line.
- ▶ Save the figure and click run.
- ▶ Now we need to start programming the call-backs.

Call-Backs

- ▶ A call-back is a function which is called when some event happens.
- ▶ When you click a push button its callback is called, when the program starts a special callback is called it is `untitled_OpeningFcn`.
- ▶ Each call-back has four parameters `hObject`, `eventdata`, `handles`, `varargin`.
- ▶ `hObject` is a handle to the figure related to the call-back.
- ▶ `eventdata` reserved to be used in the future.
- ▶ `handles` is used for user data and it is shared between all callbacks.
- ▶ `varargin` contains any command line arguments which are passed when running the figure.

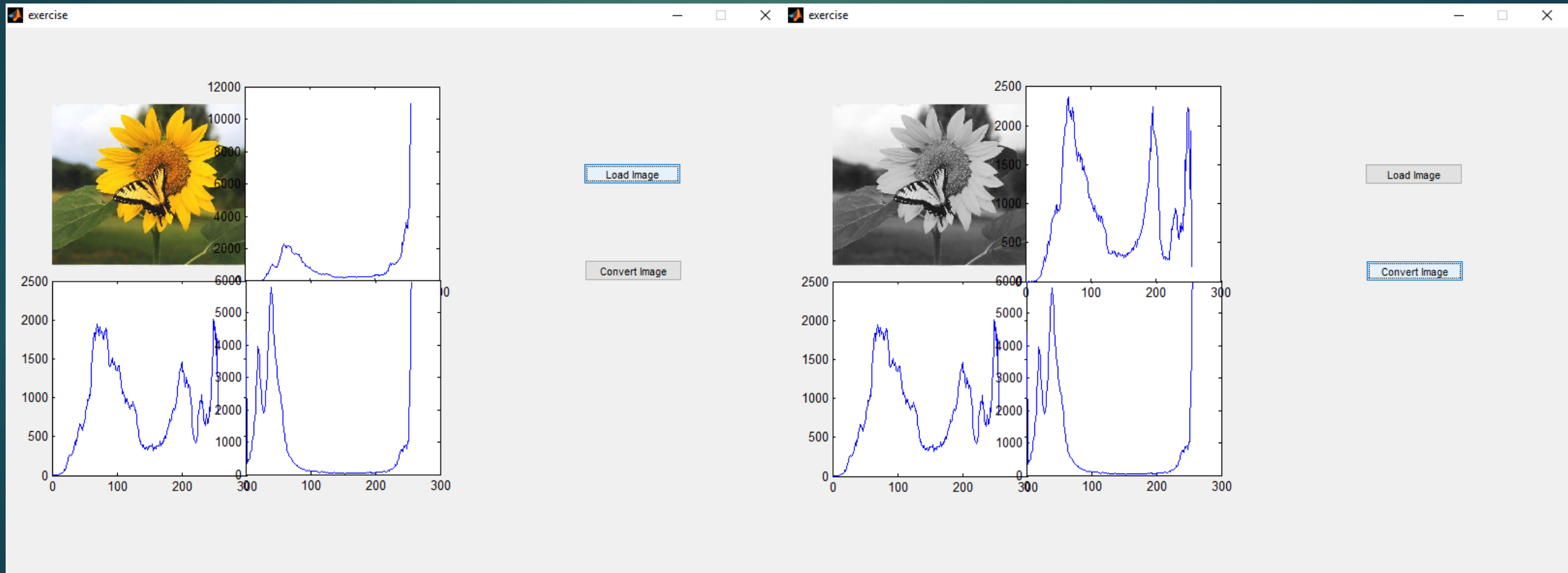
Call-Backs

- ▶ The `untitled_OpeningFcn` call back is used to populate the handles structure with data types and current data.
- ▶ The call backs for push buttons are used to plot the current data, e.g. the contour button call back has the following code `contour(handles.current_data);`
- ▶ The call back for pop-up menu contains the code which changes the current data as per selection.
- ▶ `guidata(hObject, handles)` is used to update the handles structure each time it is changed by any call back.
- ▶ Check the attachment files for more information.

Exercise

- ▶ Now up to you.
- ▶ Create a UI which has a push button to read an image.
- ▶ Display the image in an axes object and all its histograms in three more axes objects.
- ▶ Add a push button to convert the image to a gray scale image and plot its histogram.
- ▶ Refer to the next slide for screenshots of the UI.

Screenshots



More information

- ▶ To open a dialog box to select a file use: `[file_name, file_path] = uigetfile();`
- ▶ Use `strcat` to join two strings
- ▶ To select an axes for writing to it use: `axes(handles.axes1);`
- ▶ You can use try catch to handle exceptions.

GOOD LUCK