Intro writing GUI's

MATLAB

What is a GUI?

Graphical User Interface

(Aside - what is “wysiwyg”?)

MatLab provides a tool called the

Graphical User Interface Development Environment

(GUIDE)

A GUI used to create GUI's.

You can also be a masochist and write the code from scratch.

A GUI should be consistent and easily understood

(if you need the manual, there's a bug in the program or a flaw in the gui).

Provide the user with the ability to use a program without having to worry about commands to run the actual program.

Possible components of a GUI -

Pushbuttons	Menus
Sliders	Interactive Graphics
List boxes	....etc

3 Essential Parts of a GUI - 1

Graphical Components
pushbuttons, edit boxes, sliders, labels, menus, etc...

Static Components
Frames, text strings,...

Both created using the MATLAB function uicontrol.

3 Essential Parts –

2

Figures – components are contained in figures.

3

Callbacks – The functions which perform the required action when a component is "pushed".

GUIDE Properties

Allows the user to drag and drop components that he/she wants in the “layout” area of the GUI.

All “guide” GUI’s start with an opening function.

Callback is performed before user has access to GUI.

GUIDE stores GUIs in two files, which are generated the first time you save or run the GUI:

– .fig file - contains a complete description of the GUI figure layout and the components of the GUI.
Changes to this file are made in the Layout Editor

– .m file - contains the code that controls the GUI.
You program callbacks in this file using the M-file Editor.

Creating a GUI

Typical stages of creating a GUI are:

1. Designing the GUI

2. Laying out the GUI
Using the Layout Editor

3. Programming the GUI
Writing callbacks in the M-file Editor

4. Saving and Running the GUI

Assessing the Value of Your GUI

Ask yourself two basic questions when designing your GUI.

- Do the users always know where they are?
- Do they always know where to go next?

Constantly answering these two questions will help you keep in perspective the goal of your GUI.

Callback function

The "meat" of the GUI process.

Opening function is first callback in every "guide" generated GUI.

Usually used to generate data used in GUI.

Callbacks define what will happen when a figure component is selected.

You must write the callback CODE!!!!

Summary

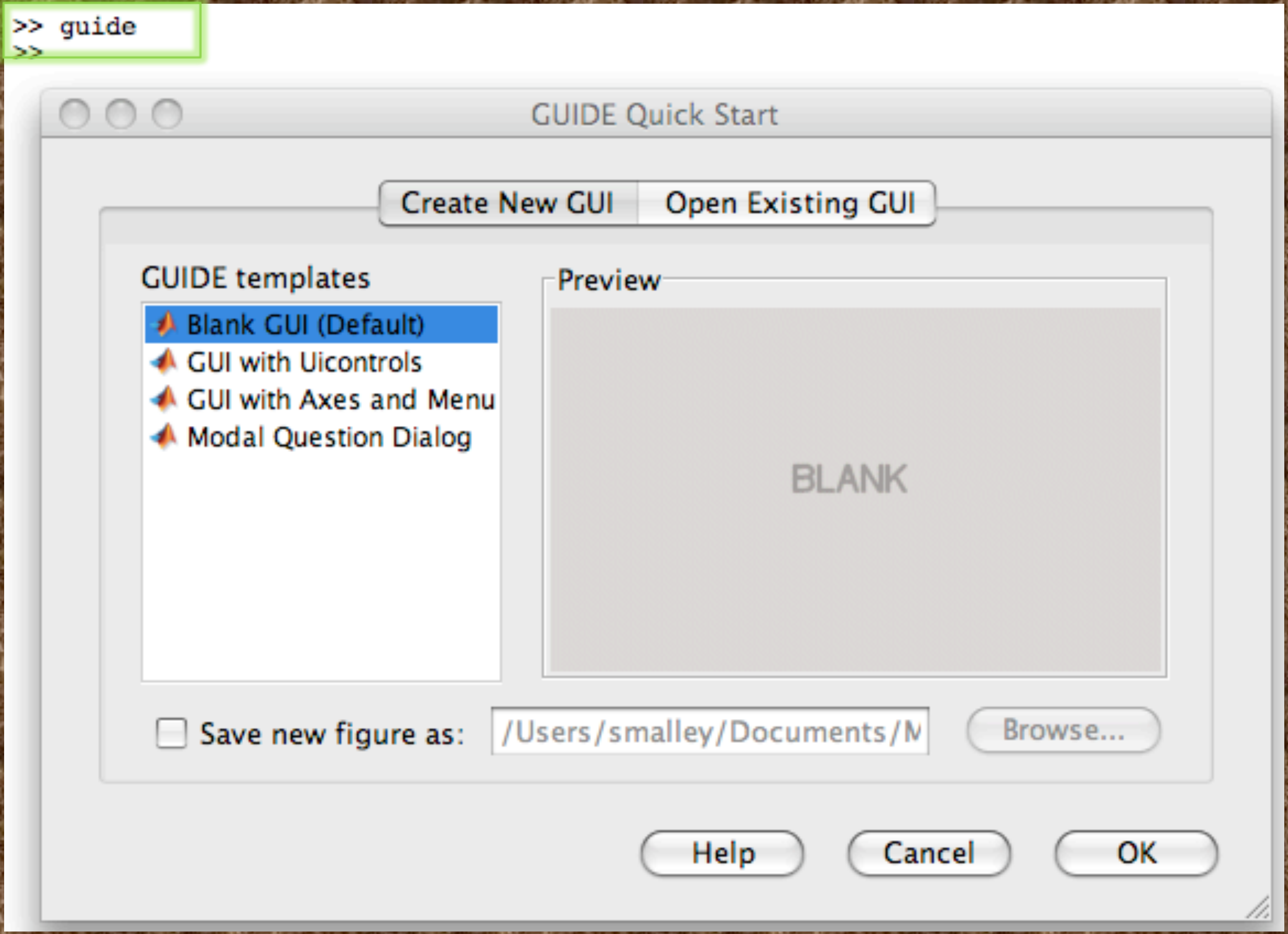
At command prompt type “guide”.

Lay out your GUI in the layout editor.

Define data in Opening Function.

Edit/Align your components using
- Tools Menu
- Align
- View menu
- Property Inspector

Write the Callbacks
(This is the most difficult aspect when creating GUI's)



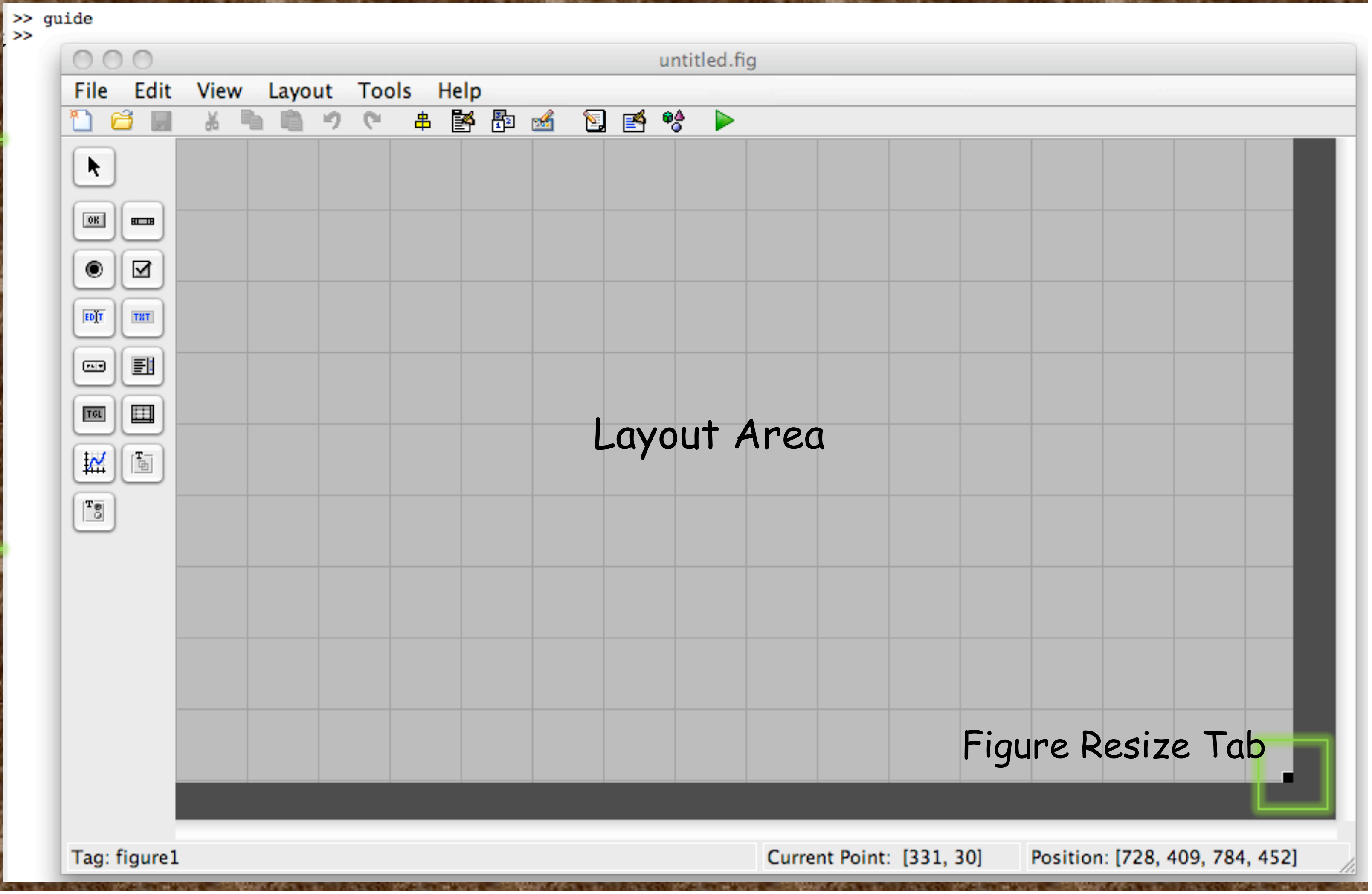
```
>> guide
>>
```

GUIDE Quick Start

Save new figure as: /Users/smalley/Documents/M

Components of GUIDE GUI interface

Alignment Tool
Menu Editor
Tab Order Editor
Toolbar Editor
M-File Editor
Property Inspector
Object Browser
Run Button


```
>> guide
>>
```

Writing Callbacks (the hard part).

A callback is a sequence of commands (function) that are execute when a graphics object is activated.

Callbacks are stored in the GUI's M-file.

Callbacks are a property of a graphics object (e.g. CreateFnc, ButtonDwnFnc, Callback, DeleteFnc).

(Also called an "event handler" in some programming languages.)

A callback is usually made of the following stages:

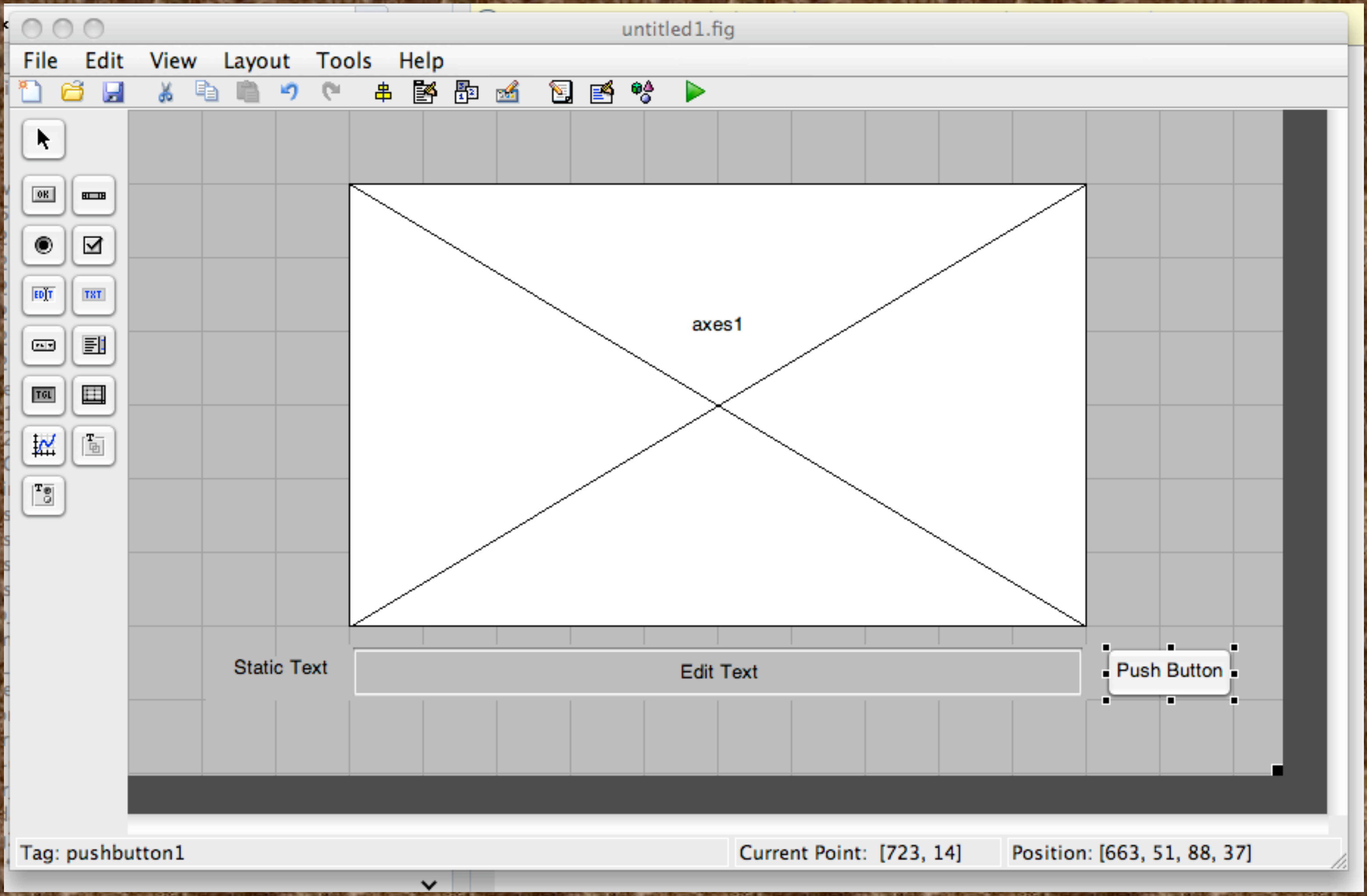
1. Get handle of object initiating the action (the object provides event / information / values).
2. Get handles of objects being affected (the object thatwhose properties are to be changed).
3. Getting necessary information / values.
4. Doing some calculations and processing.
5. Setting relevant object properties to effect action.

Let's create a GUI that plots a function that we can interactively specify.

We first lay out the basic controls for our program, selected from the menu along the left side:

axes,
static text,
edit box,
and a button.

Define and place the axis, static text (will have the prompt for the function), edit text (to interactively enter the function), and a button to do the plot.



Tag: pushbutton1
Current Point: [723, 14]
Position: [663, 51, 88, 37]

Basic Elements of our GUI-

axes: a place to draw.

static text: text that is stuck/fixed/static on the screen, the user can't edit it.

edit box: a white box that the user can type input into.

button: performs an action when user clicks on it.

The Property Inspector

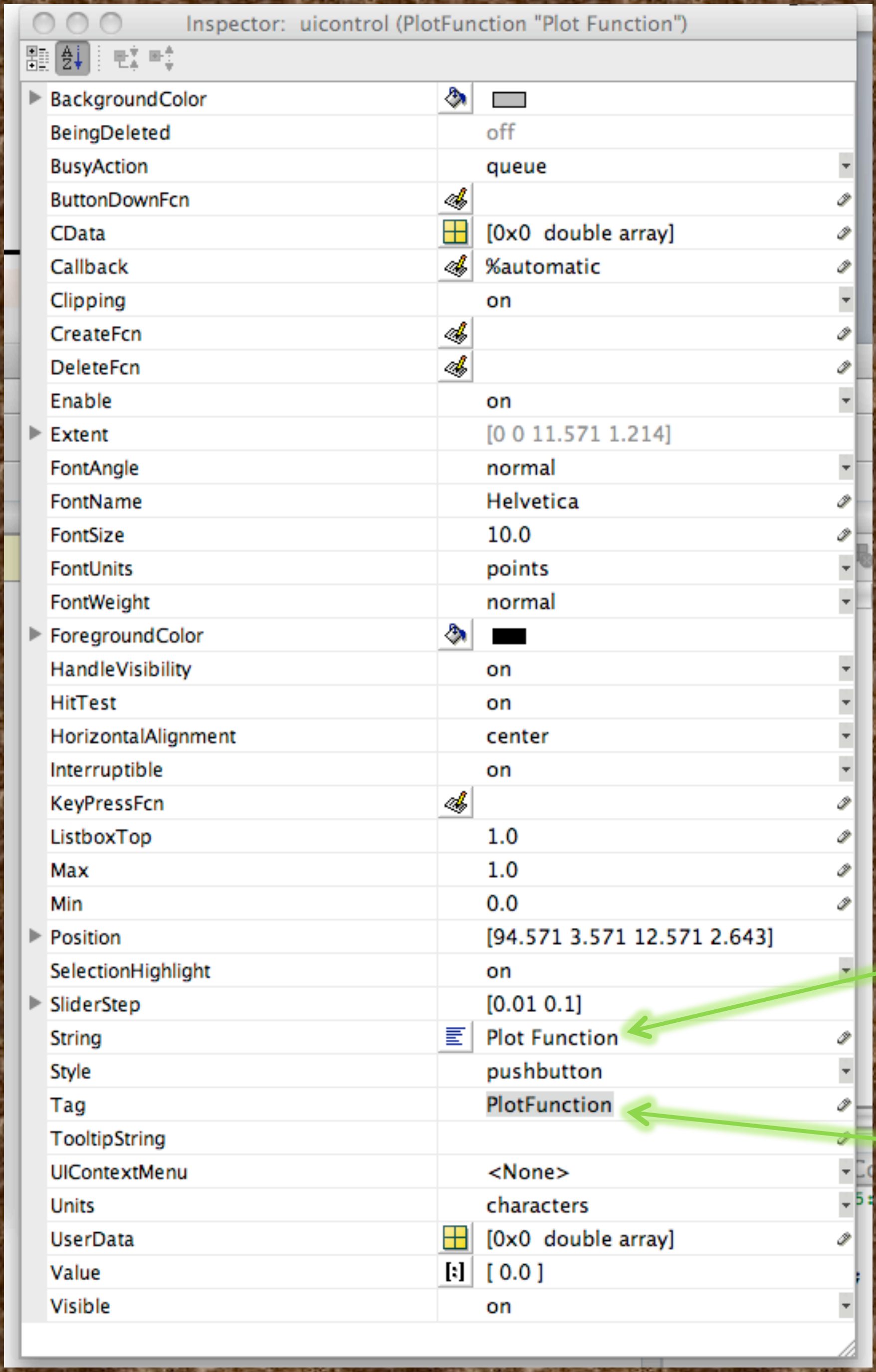
When you double-click on a control, it brings up a window listing all the properties of that control (font, position, size, etc.)

Tag - the name of the control in the code. best to rename it to something identifiable ("PlotButton" vs "button1")

String - the text that appears on the control

ForegroundColor - color of the text

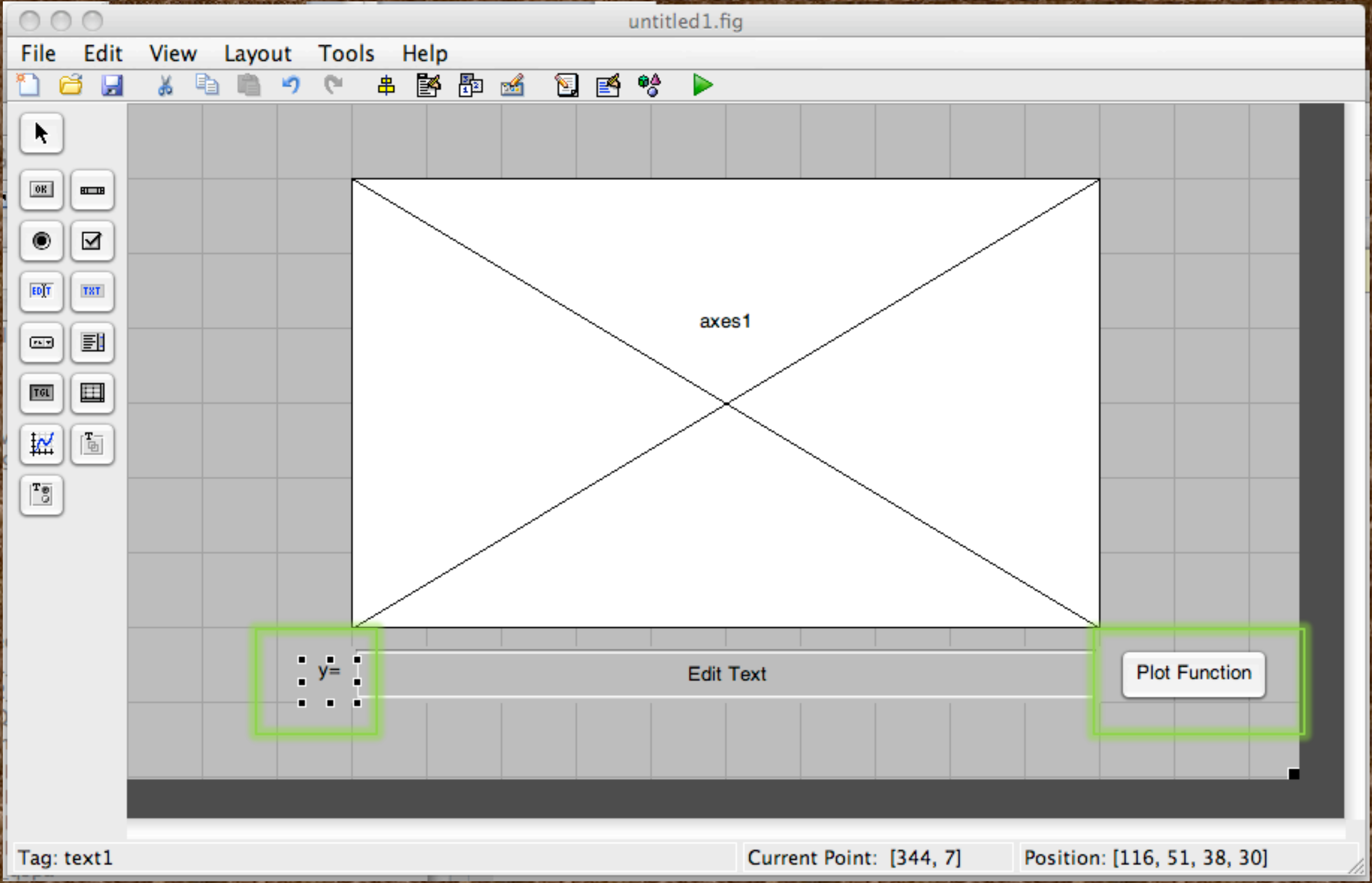
BackgroundColor - color of the control


Inspector: uicontrol (PlotFunction "Plot Function")

BackgroundColor	▭
BeingDeleted	off
BusyAction	queue
ButtonDownFcn	
CData	[0x0 double array]
Callback	%automatic
Clipping	on
CreateFcn	
DeleteFcn	
Enable	on
Extent	[0 0 11.571 1.214]
FontAngle	normal
FontName	Helvetica
FontSize	10.0
FontUnits	points
FontWeight	normal
ForegroundColor	■
HandleVisibility	on
HitTest	on
HorizontalAlignment	center
Interruptible	on
KeyPressFcn	
ListboxTop	1.0
Max	1.0
Min	0.0
Position	[94.571 3.571 12.571 2.643]
SelectionHighlight	on
SliderStep	[0.01 0.1]
String	Plot Function
Style	pushbutton
Tag	PlotFunction
TooltipString	
UIContextMenu	<None>
Units	characters
UserData	[0x0 double array]
Value	[0.0]
Visible	on

Enter text string for pushbutton

Enter tag for pushbutton



Tag: text1 Current Point: [344, 7] Position: [116, 51, 38, 30]

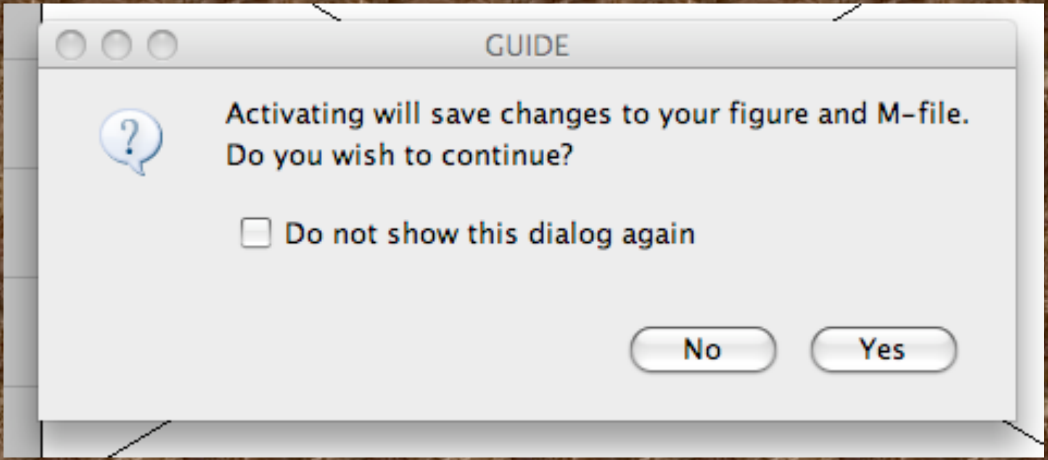
Running

If you press the green arrow at the top of the GUI editor, it will save your current version and run the program.

The first time you run it, it will ask you to name the program.

Our figure looks about right, but it doesn't do anything yet.

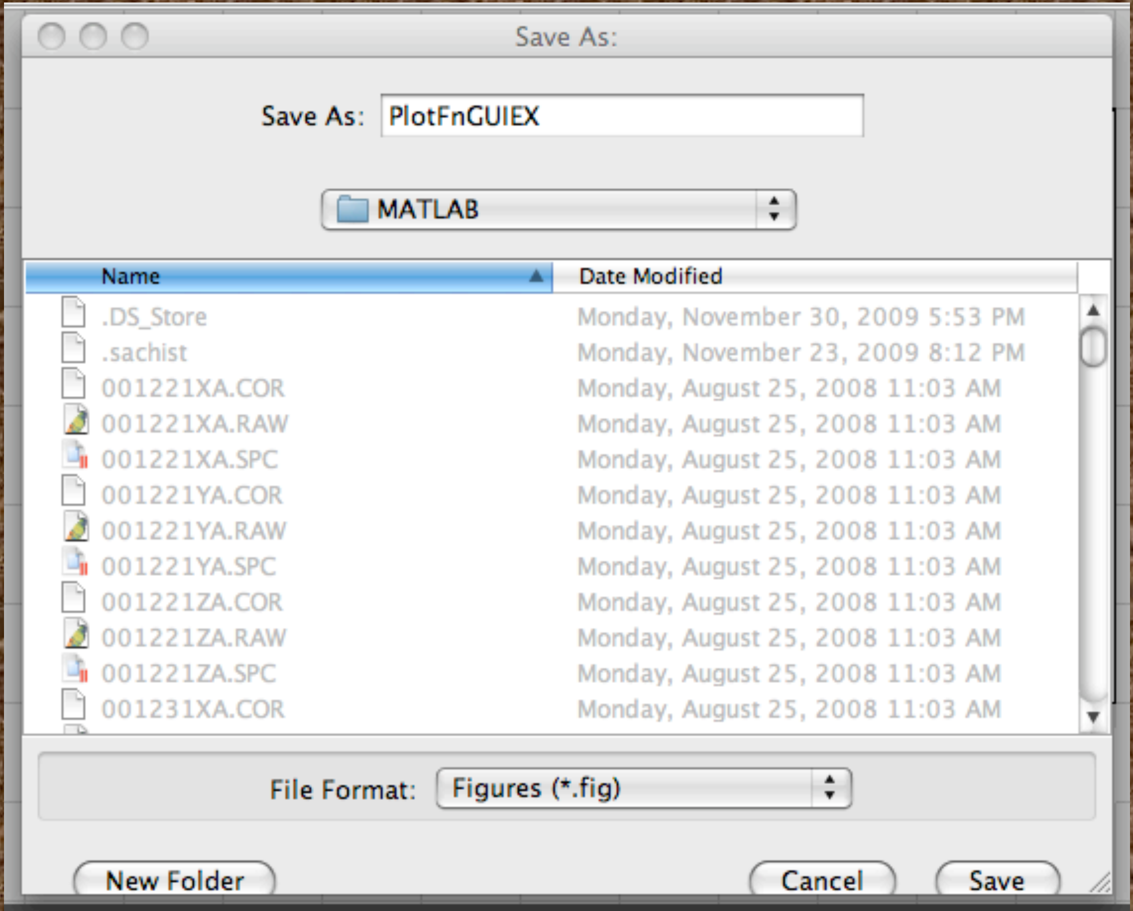
We have to define a callback for the button so it will plot the function when we press it.



GUIDE

Activating will save changes to your figure and M-file. Do you wish to continue?

☐ Do not show this dialog again



Save As:

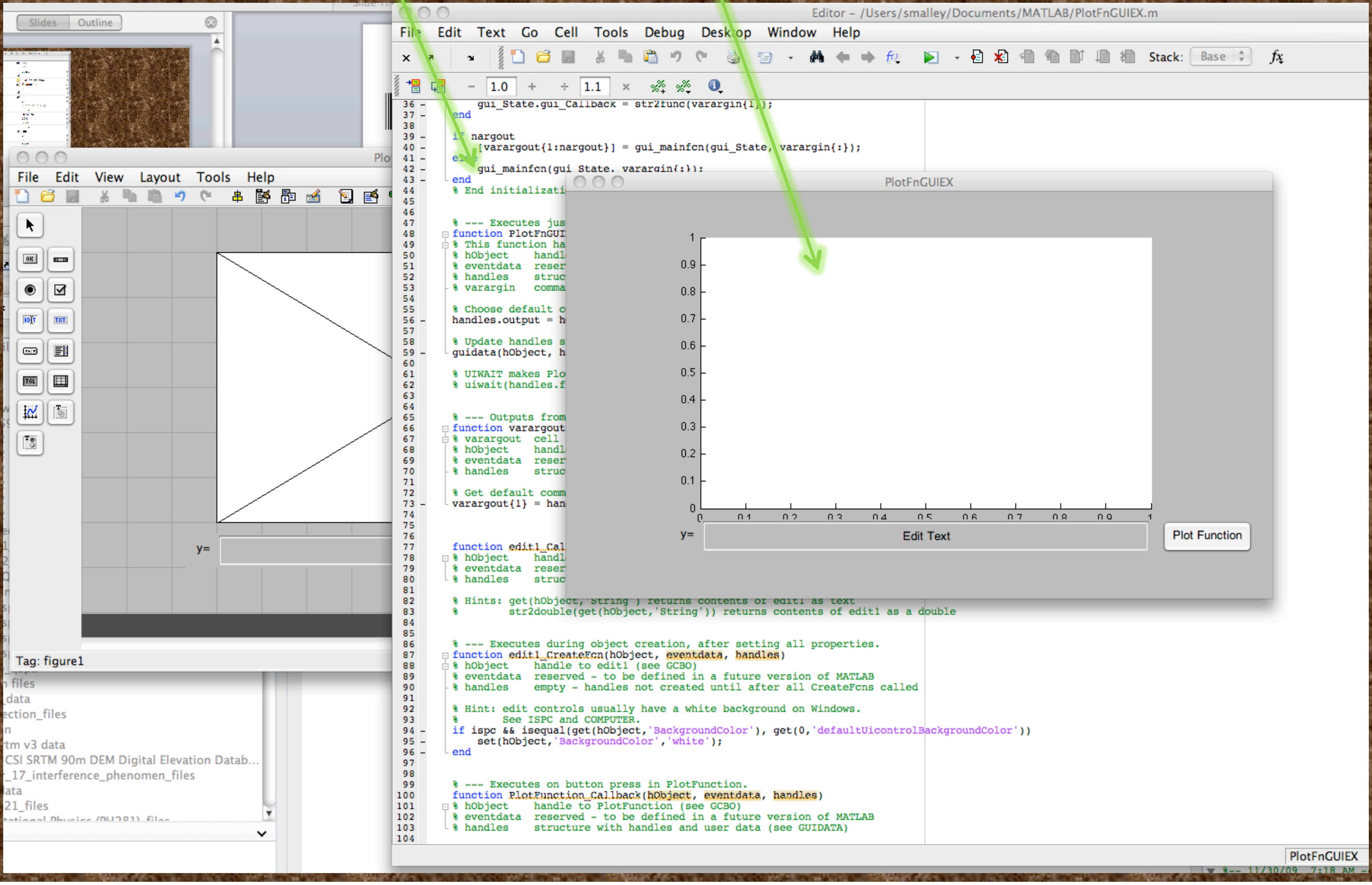
Save As: PlotFnGUIEX

MATLAB

Name	Date Modified
.DS_Store	Monday, November 30, 2009 5:53 PM
.sachist	Monday, November 23, 2009 8:12 PM
001221XA.COR	Monday, August 25, 2008 11:03 AM
001221XA.RAW	Monday, August 25, 2008 11:03 AM
001221XA.SPC	Monday, August 25, 2008 11:03 AM
001221YA.COR	Monday, August 25, 2008 11:03 AM
001221YA.RAW	Monday, August 25, 2008 11:03 AM
001221YA.SPC	Monday, August 25, 2008 11:03 AM
001221ZA.COR	Monday, August 25, 2008 11:03 AM
001221ZA.RAW	Monday, August 25, 2008 11:03 AM
001221ZA.SPC	Monday, August 25, 2008 11:03 AM
001231XA.COR	Monday, August 25, 2008 11:03 AM

File Format: Figures (*.fig)

Pile of windows - GUIDE design window, m file with code for GUI, window with running GUI.

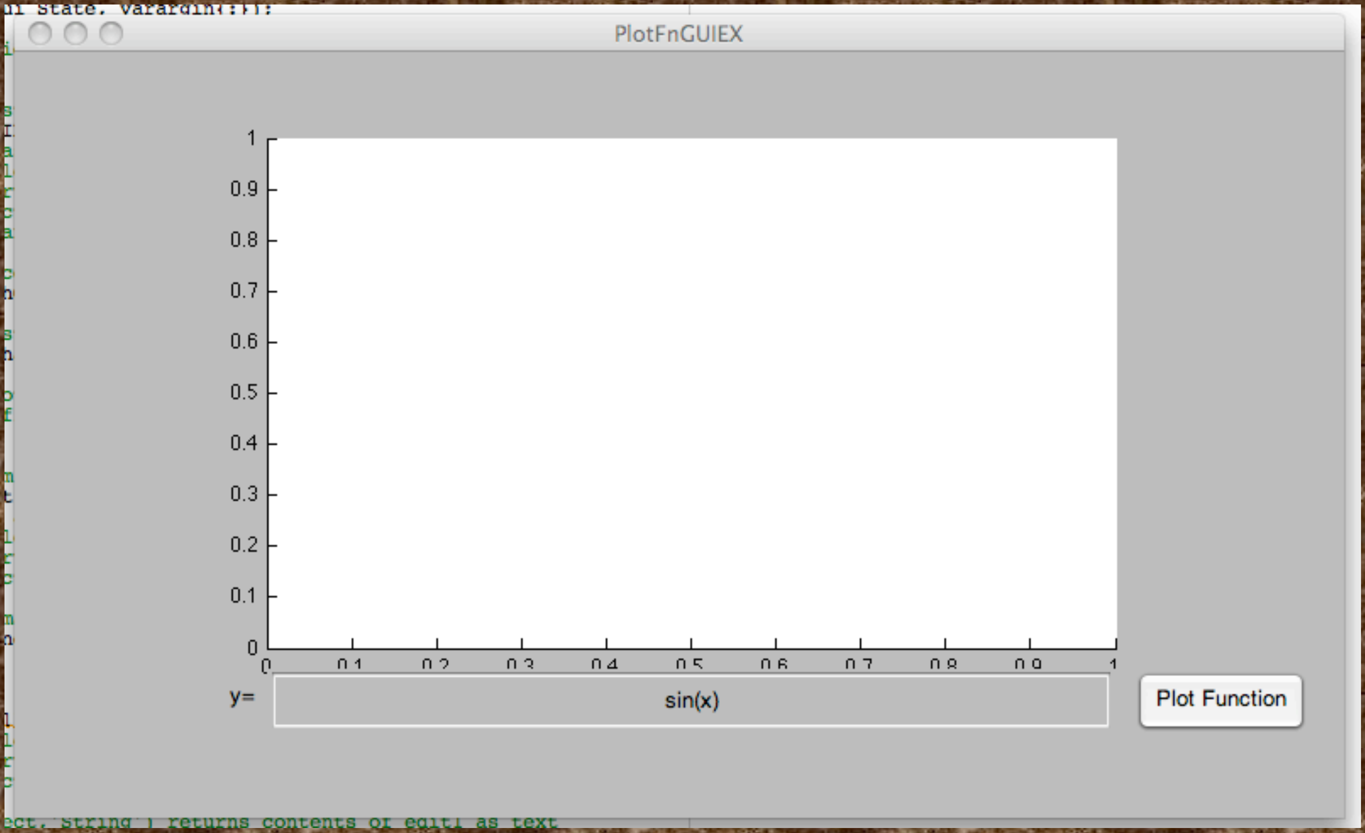

Editor - /Users/smalley/Documents/MATLAB/PlotFnGUIEX.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
File Edit View Layout Tools Help
Tag: figure1
y=
PlotFnGUIEX
Edit Text
Plot Function

```
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initializati


    % --- Executes jus
    function PlotFnGUI
    % This function ha
    % hObject    handl
    % eventdata  reser
    % handles    struc
    % varargin   comma

    % Choose default c
    handles.output = h

    % Update handles s
    guidata(hObject, h

    % UIWAIT makes Plo
    % uiwait(handles.f


    % --- Outputs from
    function varargout
    % varargout  cell
    % hObject    handl
    % eventdata  reser
    % handles    struc

    % Get default comm
    varargout{1} = han



    function edit1_Cal
    % hObject    handl
    % eventdata  reser
    % handles    struc

    % Hints: get(hObject,'String') returns contents of edit1 as text
    %        str2double(get(hObject,'String')) returns contents of edit1 as a double


    % --- Executes during object creation, after setting all properties.
    function edit1_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to edit1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %       See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end


    % --- Executes on button press in PlotFunction.
    function PlotFunction_Callback(hObject, eventdata, handles)
    % hObject    handle to PlotFunction (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
```

Buttons "work" (respond when click in them), can enter text.

But nothing happens.

Have to write callback routine to specify what happens.



Writing Callbacks

When you run the program, it creates two files.

your_gui.fig -- contains the layout of your controls

your_gui.m -- contains code that defines a callback function for each of your controls

We generally don't mess with the initialization code in the mfile.

We will probably leave many of the control callbacks blank.

Writing Callbacks

In our example, we just need to locate the function for the button.

This is why it is important to have a good Tag so we can keep our controls straight.

You can also right-click on the control and select View Callback.

Writing Callbacks

Initially the button callback looks like this.

```
% --- Executes on button press in PlotFunction.
function PlotFunction_Callback(hObject, eventdata, handles)
% hObject    handle to PlotFunction (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

We can delete the comments and type code. Note every function has the parameter handles.

This contains all the controls: handles.PlotButton, handles.edit1, handles.axes1, …

We can add variables to handles to make them available to all functions:

```
handles.x = 42;
```

Writing Callbacks

We can look up any property of a control with the get function.

Similarly, we can change any property with the set function.

This is where things get complicated.

Writing Callbacks

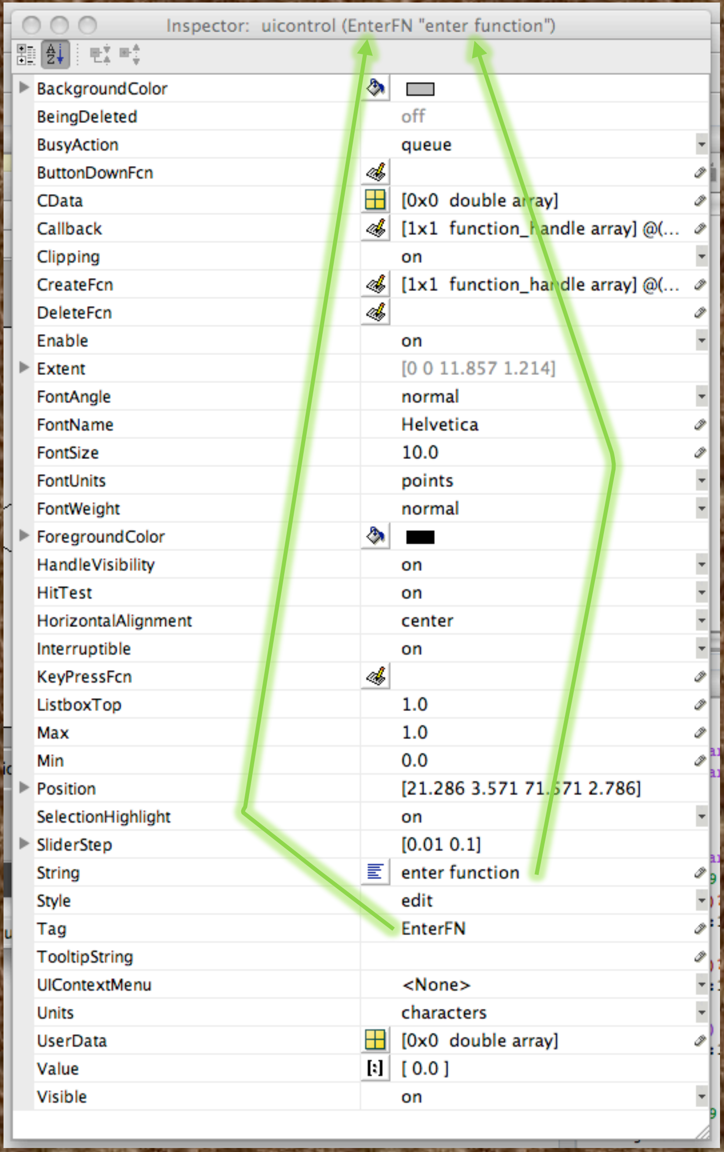
We need two callbacks.

1) We want to get the String typed into the edit box

2) and plot it.

```
function EnterFN_Callback(hObject, eventdata, handles)
. . .
function EnterFN_CreateFcn(hObject, eventdata, handles)
```

Look at properties inspector and m file to see how things match up.



Inspector: uicontrol (EnterFN "enter function")

BackgroundColor	▭
BeingDeleted	off
BusyAction	queue
ButtonDownFcn	
CData	[0x0 double array]
Callback	[1x1 function_handle array] @(...
Clipping	on
CreateFcn	[1x1 function_handle array] @(...
DeleteFcn	
Enable	on
Extent	[0 0 11.857 1.214]
FontAngle	normal
FontName	Helvetica
FontSize	10.0
FontUnits	points
FontWeight	normal
ForegroundColor	■
HandleVisibility	on
HitTest	on
HorizontalAlignment	center
Interruptible	on
KeyPressFcn	
ListboxTop	1.0
Max	1.0
Min	0.0
Position	[21.286 3.571 71.571 2.786]
SelectionHighlight	on
SliderStep	[0.01 0.1]
String	enter function
Style	edit
Tag	EnterFN
TooltipString	
UIContextMenu	<None>
Units	characters
UserData	[0x0 double array]
Value	[0.0]
Visible	on

1) We want to get the string typed into the edit box

Cyan produced by guide, have to add the white. Variable <u>`handles.EnterFn`</u> created here.

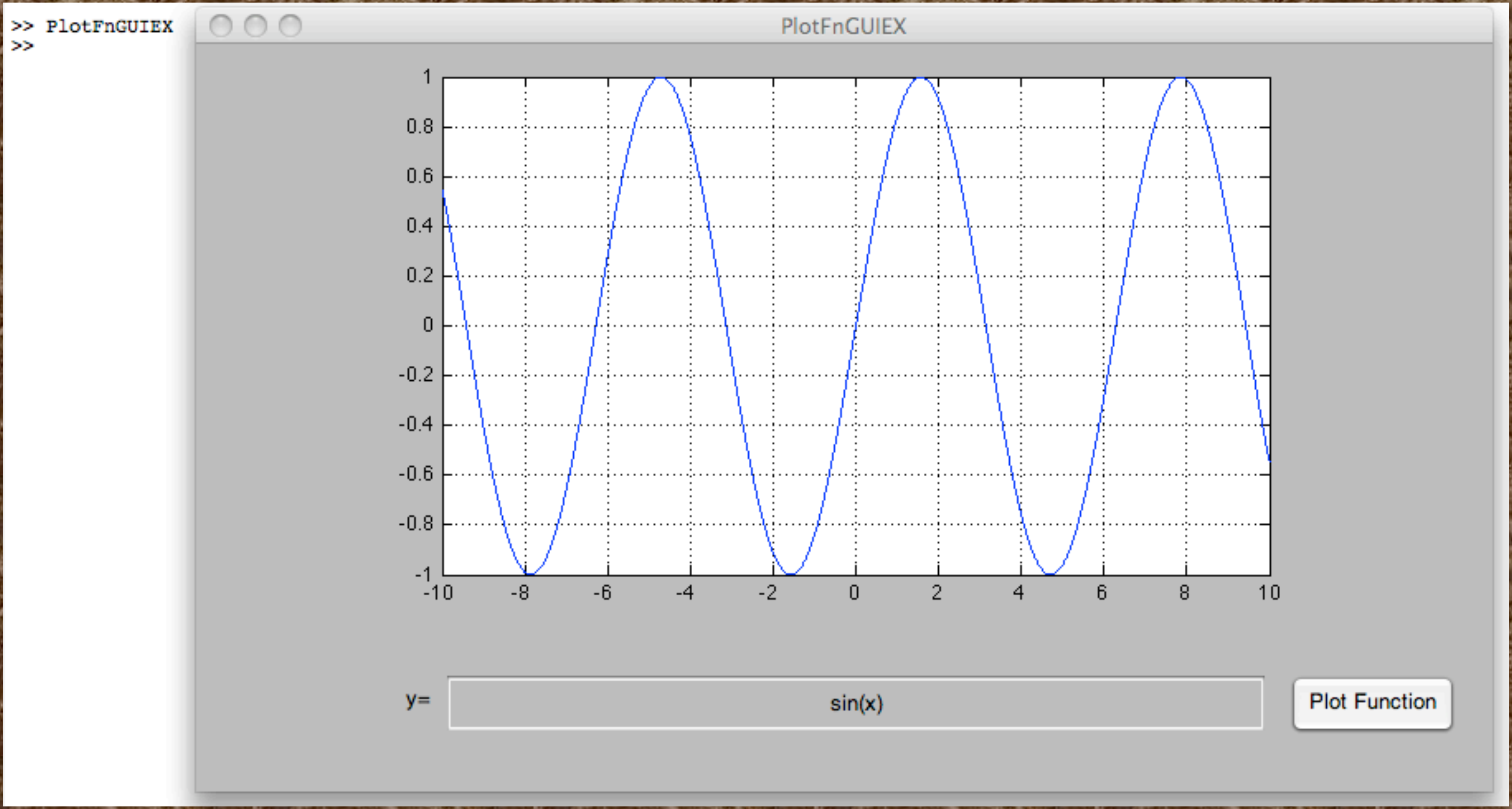
```
function EnterFN_Callback(hObject, eventdata, handles)
% hObject    handle to EnterFN (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of EnterFN as
text
%        str2double(get(hObject,'String')) returns contents of
EnterFN as a double
handles.EnterFn=get(hObject,'String');
```

2) and plot it.

Cyan produced by guide, have to add the white. Variable `handles.EnterFn` created by us, while `handles.axes1` created by guide.

```
% --- Executes on button press in PlotFunction.
function PlotFunction_Callback(hObject, eventdata, handles)
% hObject    handle to PlotFunction (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
x=-10:.01:10
s = get(handles.EnterFN, 'String');
y = eval(s); %eval just evaluates the given string
handles.axes1; %Subsequent commands draw on axes1.
plot(x, y);
grid;
```

Final result.



```
>> PlotFnGUIEX
>>
```

y= sin(x)