My First Matlab GUI (Version 1.1)
BE-205, Winter '06-'07, Dr. C. S. Tritt

Background

This simple program initially displays “Unclicked” in a static text field and two buttons (“Click Me” and “Exit”). After the “Click Me” button is clicked, the message changes to “Clicked!” Clicking the “Exit” button closes the program.

Interface Design (with Tags)

A static text field (theText) initially saying “Unclicked.”

A button (theButton) saying “Click Me.”

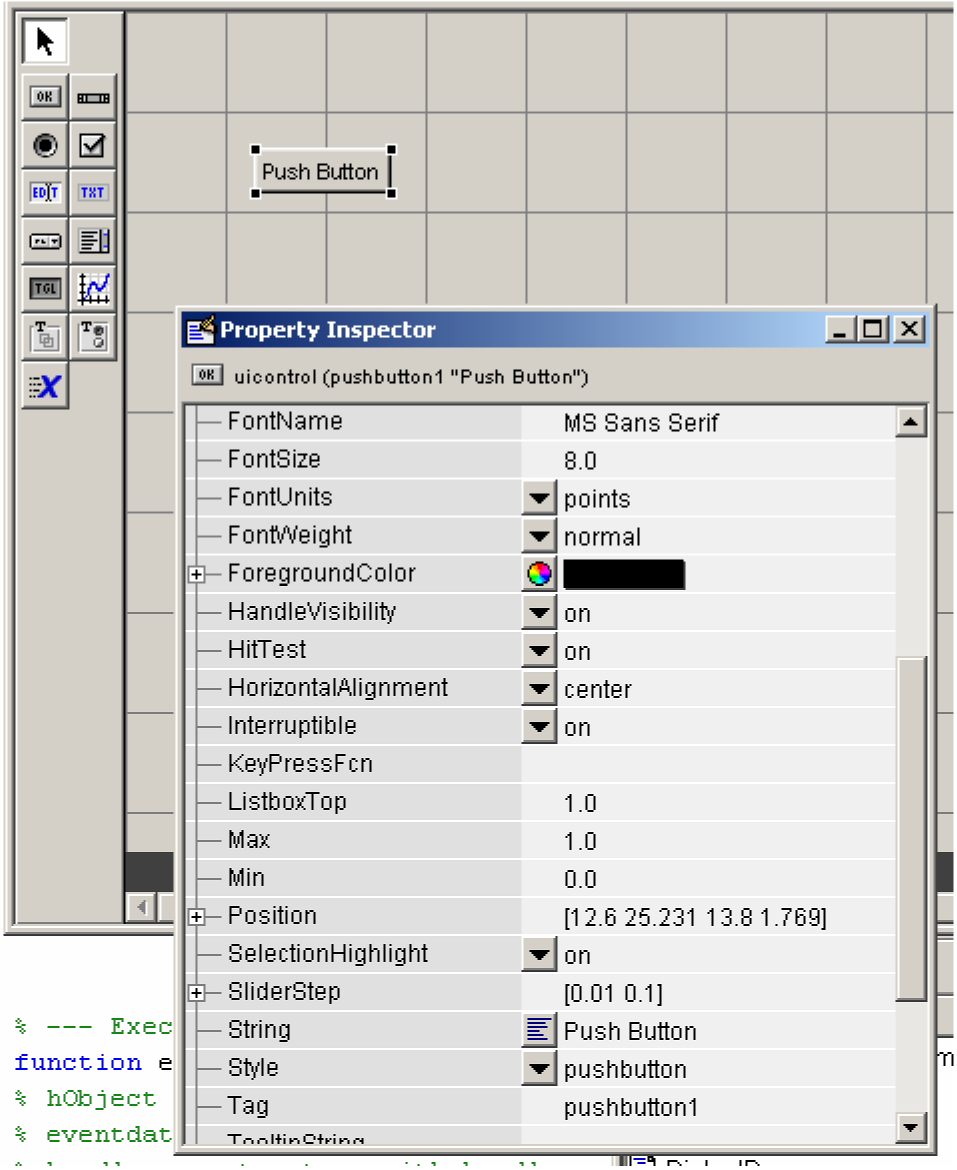
A button (exitButton) saying “Exit.”

Interface Creation

Use GUIDE to create a Blank GUI .

Position the static text and buttons by dragging them from the toolbar. Adjust the size of the GUI figure and its contents (controls). Note that adjusting the figure size automatically moves and resizes its contents.

Double click on each control to display the Property Inspector. The *FontSize*, *String* and *Tag* properties are particularly important. The *FontSize* property controls the size of any displayed text. The *String* property is the text displayed and the *Tag* property is the name of the control. The *Tag* is used by GUIDE to generate callback function names and by the programmer to access the control via the handles structure.

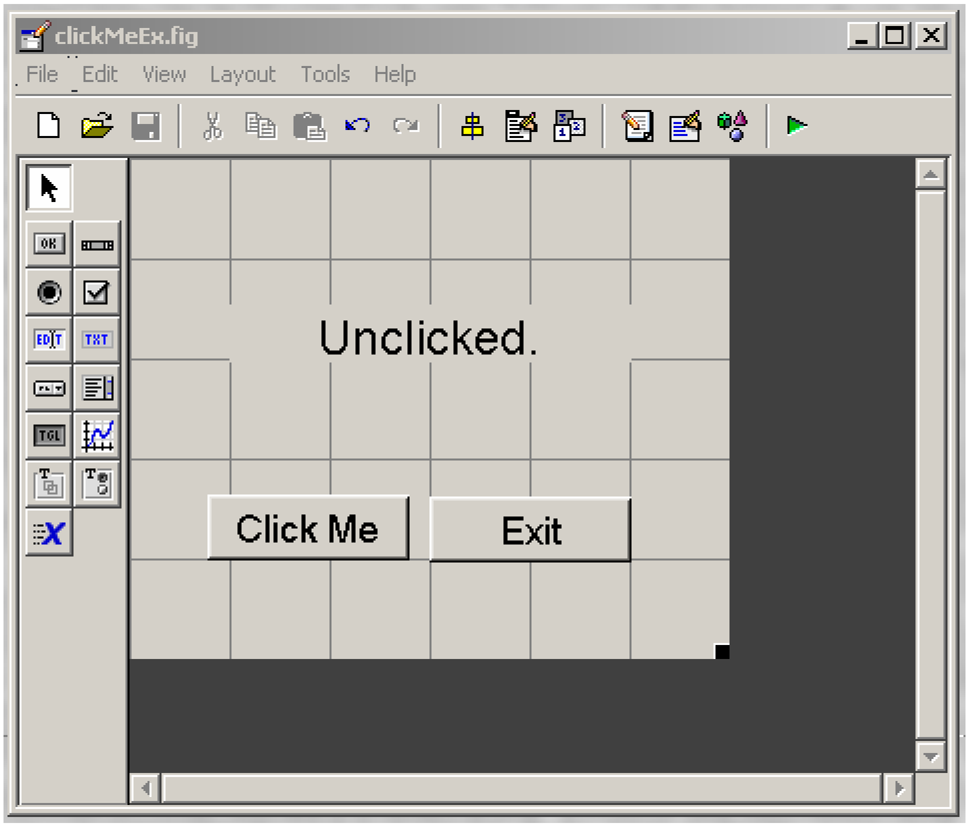


uicontrol (pushbutton1 "Push Button")

FontName	MS Sans Serif
FontSize	8.0
FontUnits	points
FontWeight	normal
ForegroundColor	■
HandleVisibility	on
HitTest	on
HorizontalAlignment	center
Interruptible	on
KeyPressFcn	
ListboxTop	1.0
Max	1.0
Min	0.0
Position	[12.6 25.231 13.8 1.769]
SelectionHighlight	on
SliderStep	[0.01 0.1]
String	Push Button
Style	pushbutton
Tag	pushbutton1
TooltipString	

```
% --- Exec
function e
% hObject
% eventdat
```

Change the font size to 14 or 16 and the strings and tags of the controls to match those listed above.



Save the figure as *clickMeEx* using *File* | *Save*.

Adding Functionality

When the figure is saved, GUIDE creates some default code for the program an .m file having the same name as the figure.

Note that GUIDE places multiple functions in a single file and omits the end statements between them. While this is different from the style you have been taught, it is perfectly legal Matlab. However, Matlab does not allow some functions to have end statements and others not to in the same file.

While GUIDE seems to be able to add additional code if you add controls to your figures, I don't recommend you use this feature unless absolutely necessary (I've seen it not work in subtle ways in Microsoft Visual Studio (C++) and Eclipse (Java)).

Now modify the generated code to add the desired behaviors.

Add the following line to the `theButton_Callback(hObject, eventdata, handles)` function:

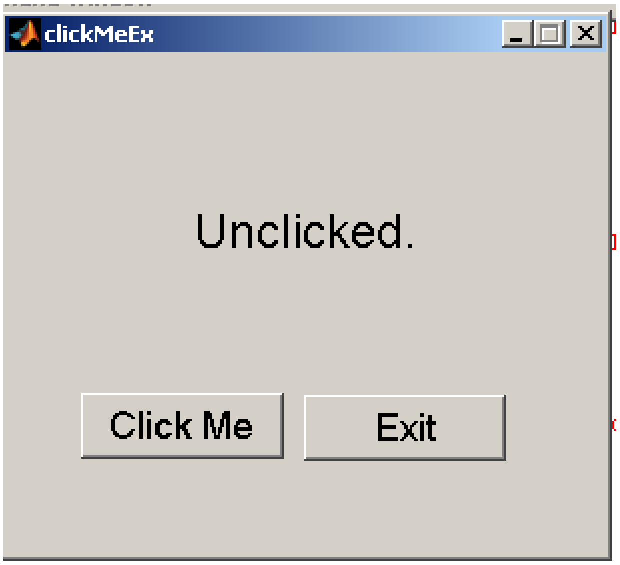
```
set(handles.theText, 'String', 'Clicked!');
```

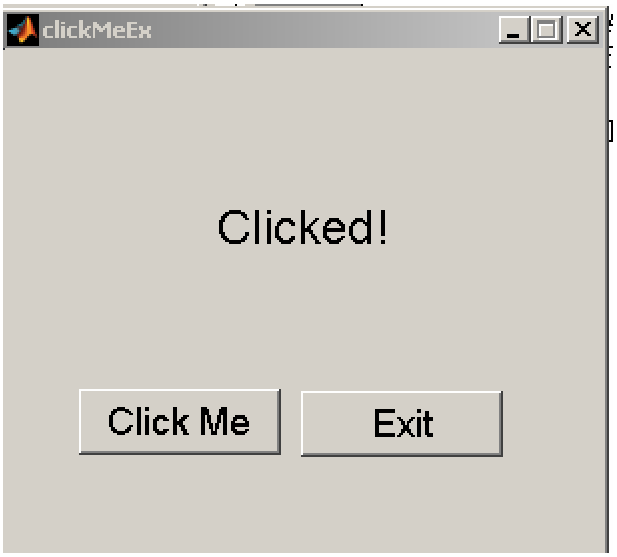
And add the following line to the `exitButton_Callback(hObject, eventdata, handles)` function:

```
delete(handles.figure1);
```

After saving these changes, go to the Matlab console window and enter the figure and .m file name (clickMeEx) to run the program.

The results are shown below:





Source Code

```
function varargout = clickMeEx(varargin)
% CLICKMEEX M-file for clickMeEx.fig
%
%   My first GUI example. Click the button to see the message change.
%   Dr. C. S. Tritt
%   1/7/07
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help clickMeEx

% Last Modified by GUIDE v2.5 07-Jan-2007 17:07:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @clickMeEx_OpeningFcn, ...
                   'gui_OutputFcn',  @clickMeEx_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    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 initialization code - DO NOT EDIT
```

Create GUI state structure to be passed to gui_mainfun.

Activate (run & display) the GUI.

```
% --- Executes just before clickMeEx is made visible.
function clickMeEx_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to clickMeEx (see VARARGIN)

% Choose default command line output for clickMeEx
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes clickMeEx wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

Initialization code goes here.

```
% --- Outputs from this function are returned to the command line.
function varargout = clickMeEx_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

Callbacks start here. Enter code to create desired functionality in them.

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

% Added this line to create functionality - cst.
set(handles.theText, 'String', 'Clicked!', 'BackgroundColor', 'r');


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

% Added this line to create functionality - cst.
delete(handles.figure1);
```

Additional clean-up code can go here.