



Matlab编程与应用

第五讲



本讲内容

- part1: GUI介绍
- Part2: 基于AppDesigner的GUI设计
- Part3: GUI的编程实现



part1

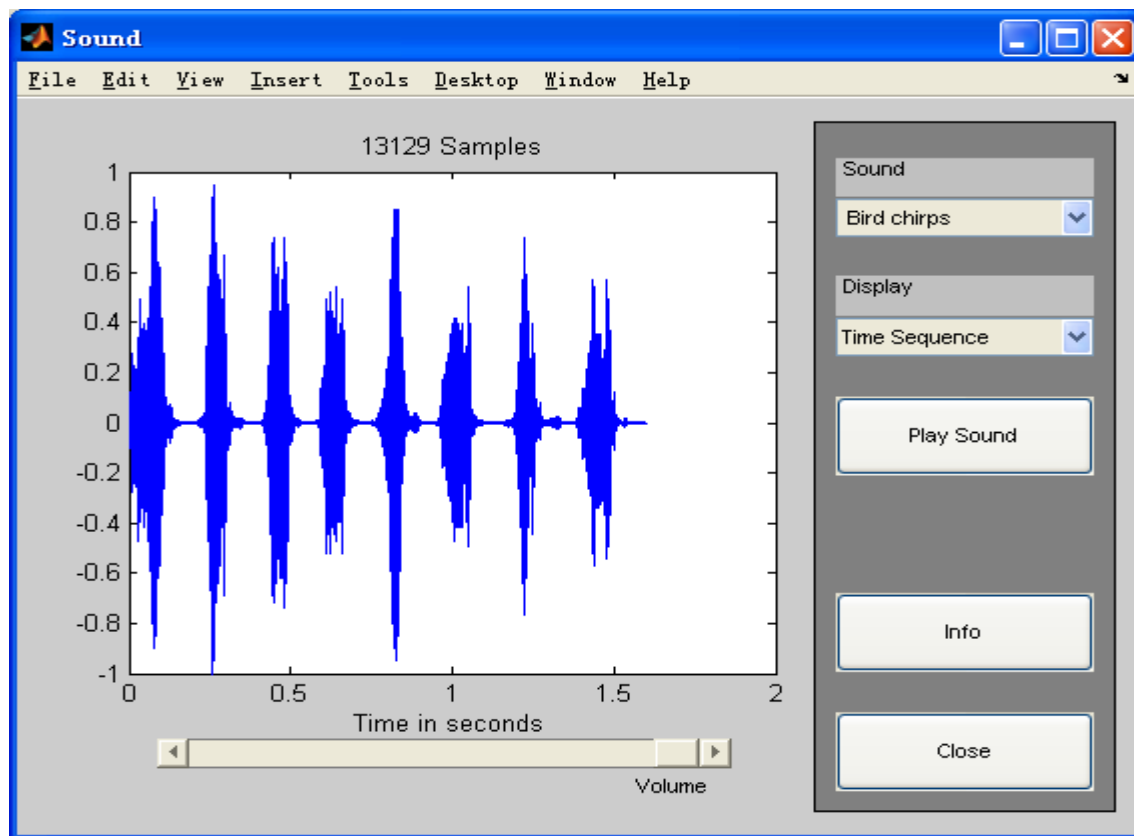
GUI介绍

GUI介绍

- GUI: (Graphical User Interfaces)图形用户界面

例：matlab的一个demo

xpsound

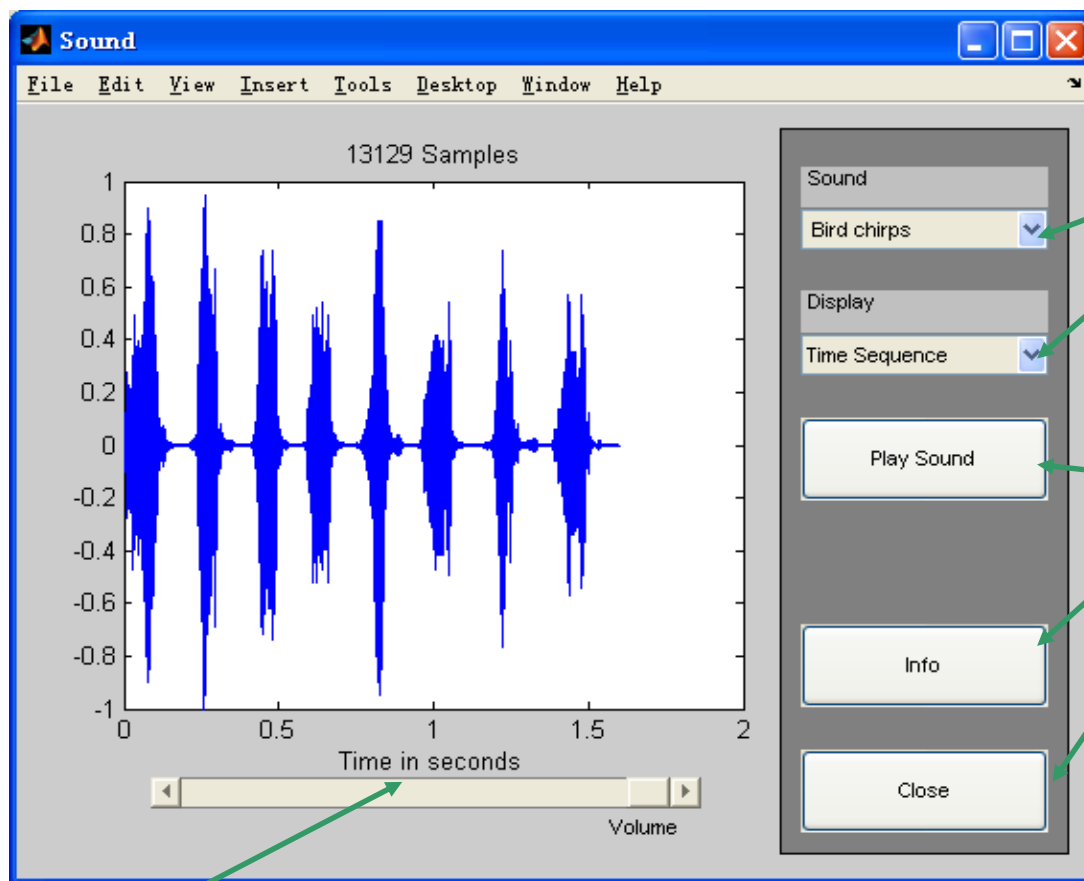




GUI介绍

- 主要利用各种控件（**uicontrols**）与用户交互： 按钮、 滑动条、 列表框...
- 每个控件也是图形对象， 也拥有属性与回调函数
- 改变控件的属性可以调整控件的外观。
利用回调函数可以对用户的操作做出相应的响应。（事件驱动）

GUI介绍



弹出式菜单
pop-up menu

按钮
push button

滑动条
slider



GUI介绍

- Matlab中创建GUI的三种方式:
 - 1.编程实现
 - 2.利用GUIDE工具
 3. 利用App Designer 工具



GUIDE

- **Graphic User Interface Designer Environment**
- **GUIDE生成两个文件：**
 - .fig文件：**保存GUI的layout,就是展现给用户的界面，界面含有axes,button,listbox等控件。
 - .m文件：**包含对各种事件的响应，即回调函数。

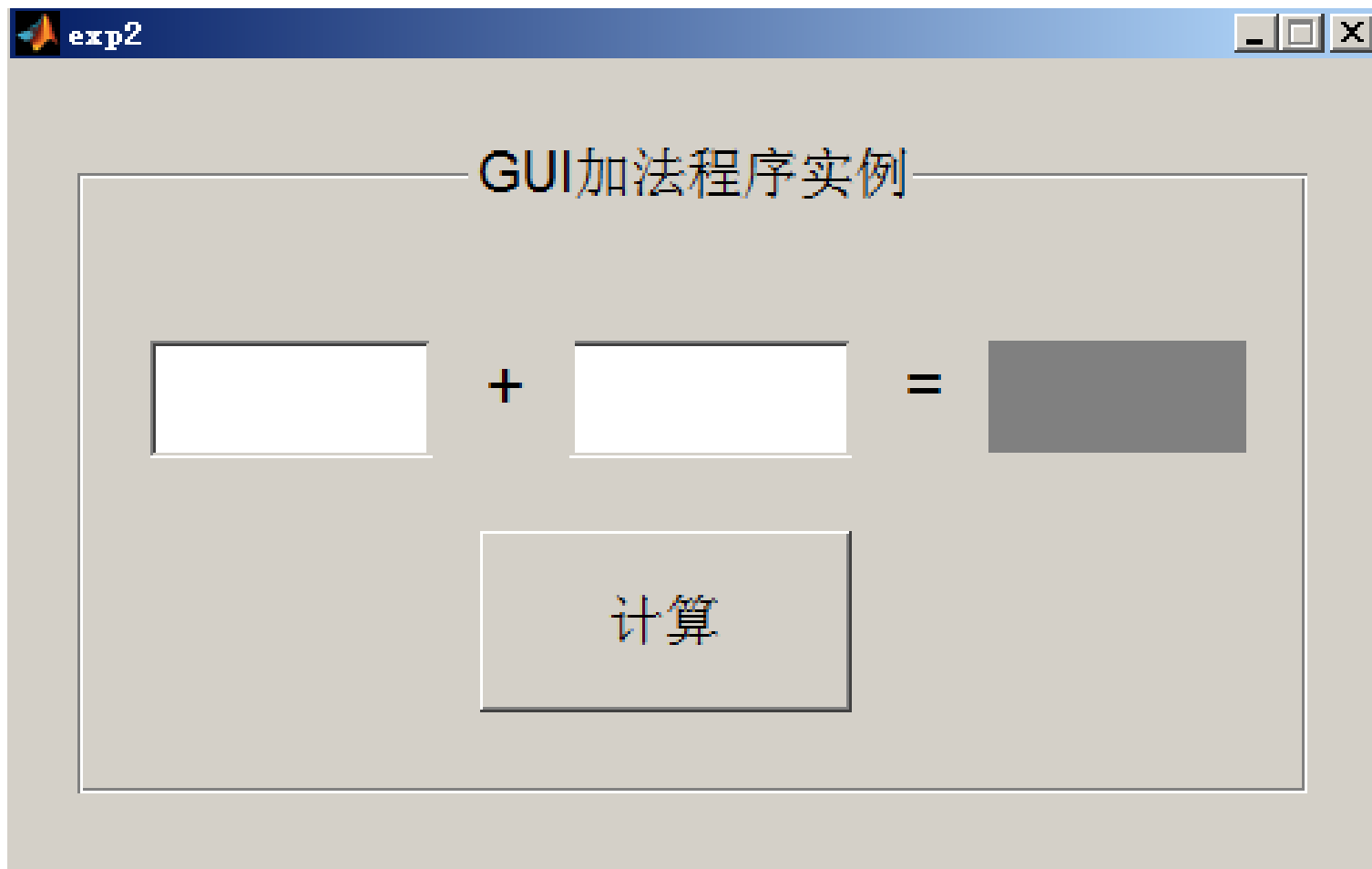


GUIDE

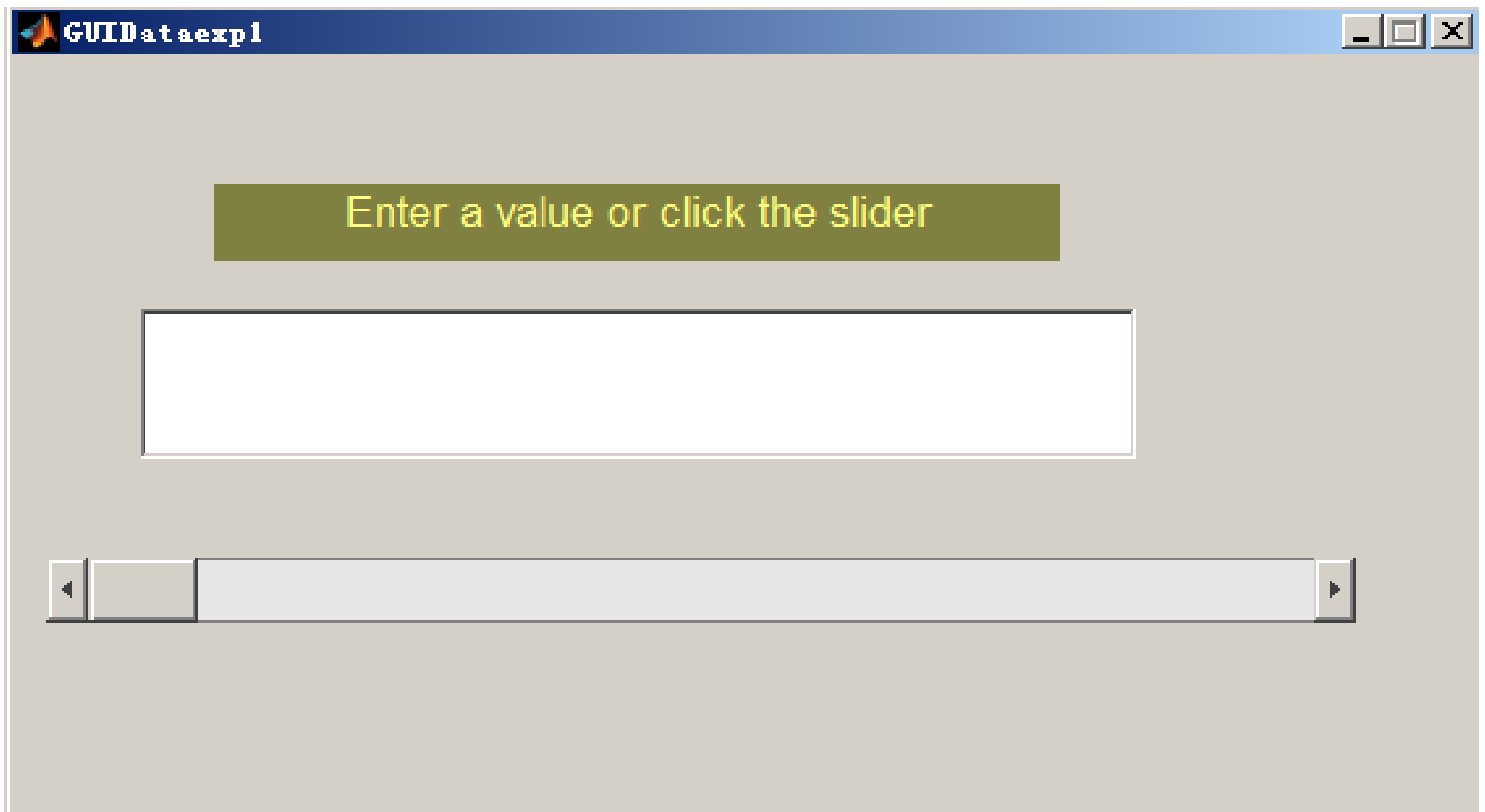
- 主要设计过程

- 1.构思整个**GUI**的布局与设计任务;
- 2.利用**GUIDE**进行界面设计;
- 3.设置界面上各个控件的属性;
- 4.编写回调函数代码。

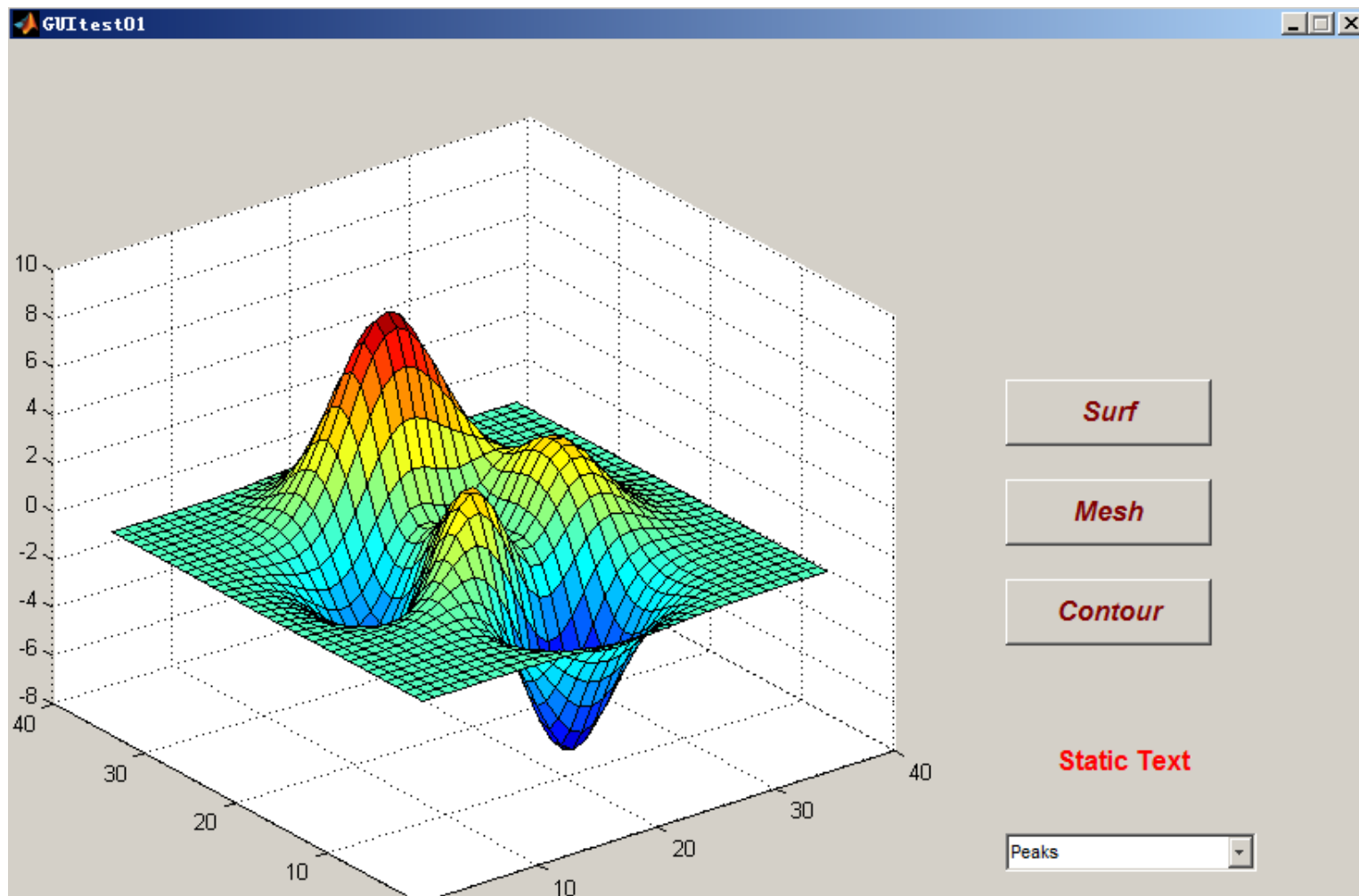
GUIDE例1



GUIDE例2

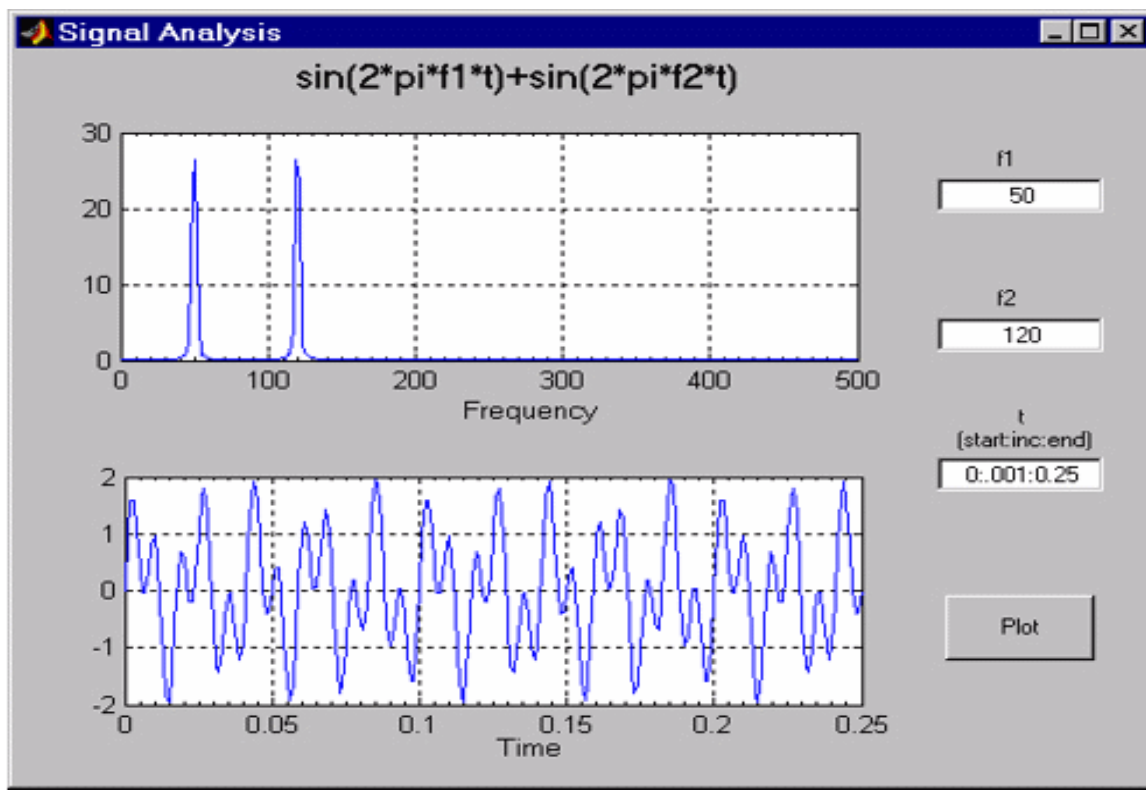


GUIDE例3



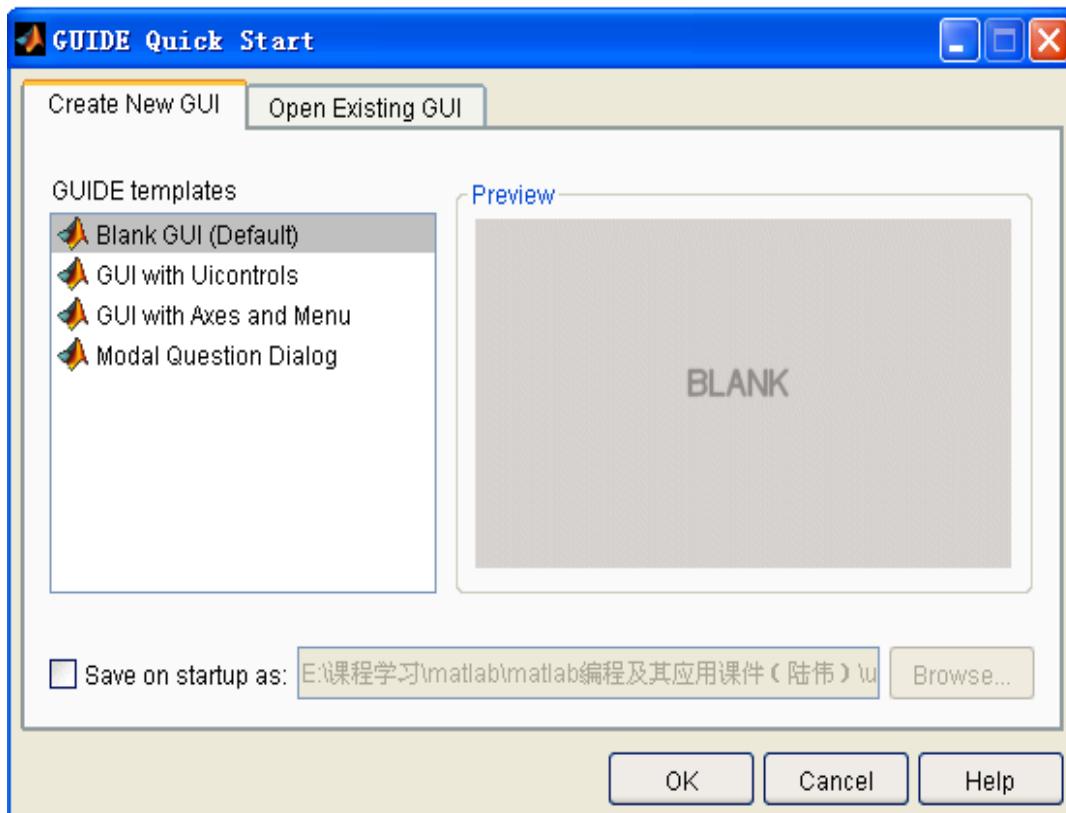
GUIDE例4

例：设计一个**GUI**，生成一个由两个不同频率正弦波相加的信号，显示其时域波形与频谱，频率值和时间可由用户输入。



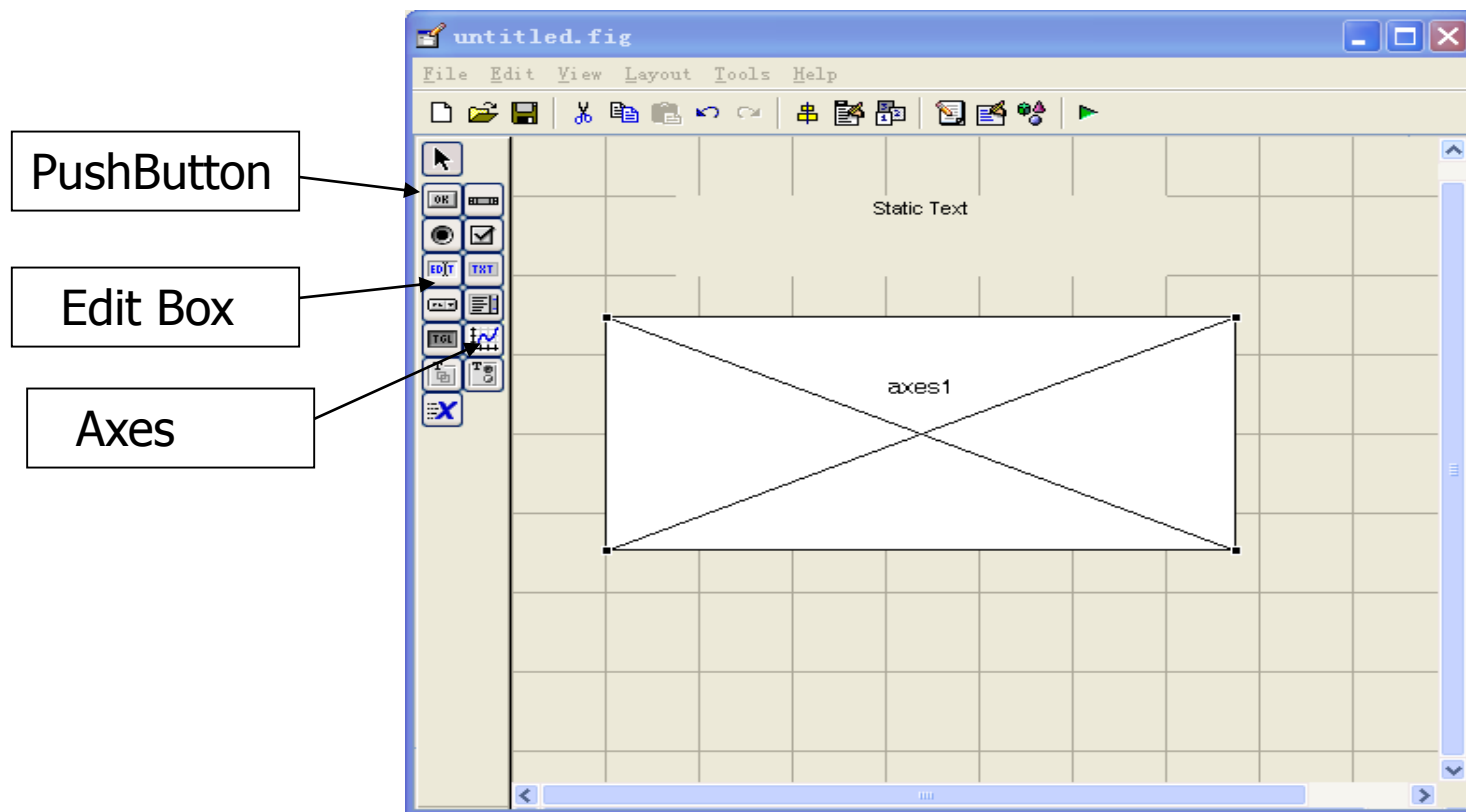
GUIDE

1.新建一个空白的GUI并保存。



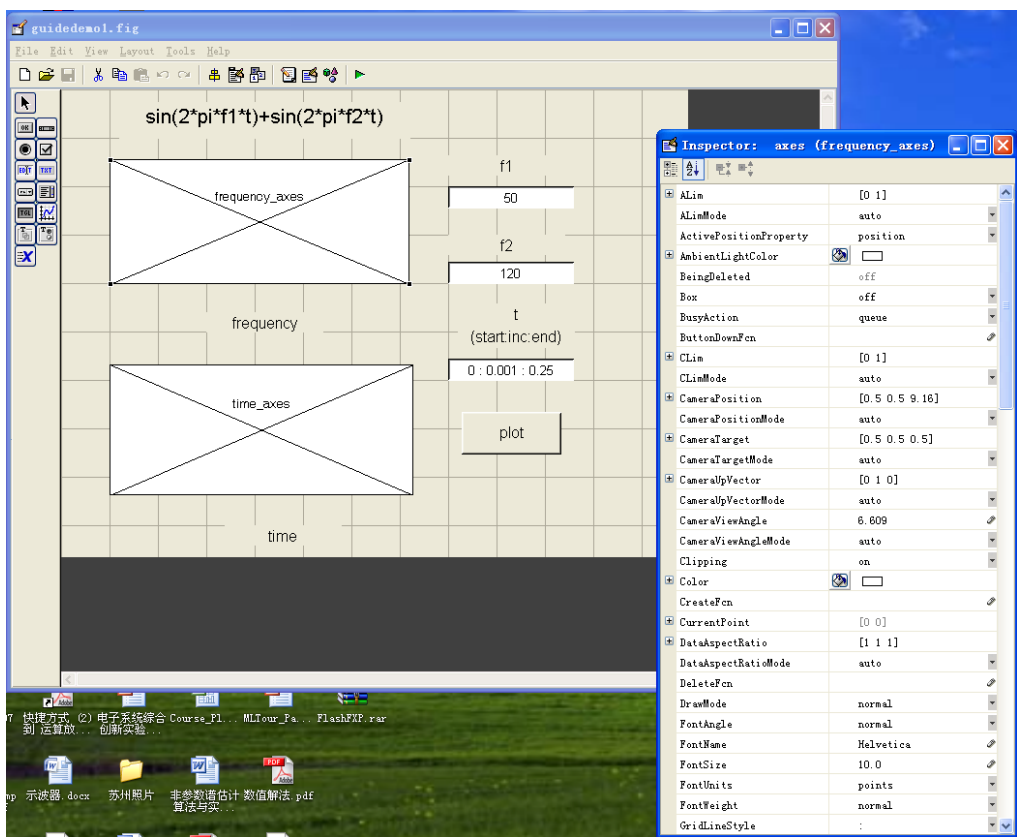
GUIDE

2.按照预先设计，在界面合适位置放置控件。



GUIDE

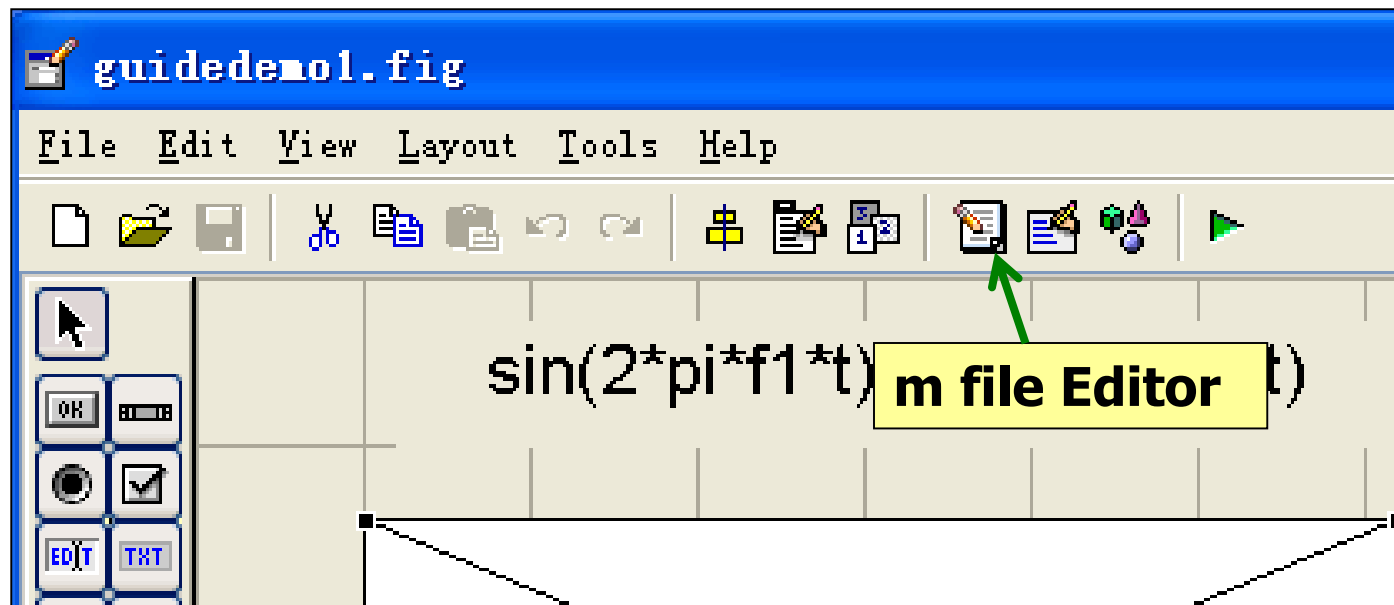
3. 双击某控件，设置控件的一些属性。



- 要恰当定义每个控件的 'tag' 属性，便于标识，如 'time_axes', 'freq_axes'。
- 每个控件 'tag' 属性不能相同。因为m文件的回调函数利用该属性找到相应的控件。

GUIDE

4. 打开对应的.m文件编写响应的回调函数





GUIDE

- m文件中有很多子函数，大部分函数不需要添加代码，但也**不能随意删除**！
- 回调函数三个参数的含义：

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% Get user input from GUI
```

GUIDE

function p
handles)

% hObject
hObject即

% eventdata
% handles
件的句柄，
字段值为该

```
Editor - E:\课程学习\matlab\matlab编程及其应用课件(陆伟)\图形句柄与GUI编程\...
File Edit Text Go Cell Tools Debug Desktop Window Help
pushbutton1_...

145 % --- Executes on button press in pushbutton1.
146 function pushbutton1_Callback(hObject, eventdata, handles)
147 % hObject handle to pushbutton1 (see GCBO)
148 % eventdata reserved - to be defined in a future version
149 % handles structure with handles and user data (see
150 % Get user input from GUI
151 f1 = str2double(get(handles.f1_input,'String'));
152 f2 = str2double(get(handles.f2_input,'String'));
153 t = eval(get(handles.t_input,'String'));
154
155 % Calculate data
156 x = sin(2*pi*f1*t) + sin(2*pi*f2*t);
157 y = fft(x,512);
158 m = y.*conj(y)/512;
159 f = 1000*(0:256)/512;
160
161 % Create frequency plot
162 axes(handles.frequency_axes)
163 plot(f,m(1:257))
164 set(handles.frequency_axes,'XMinorTick','on')
165 grid on
166
```

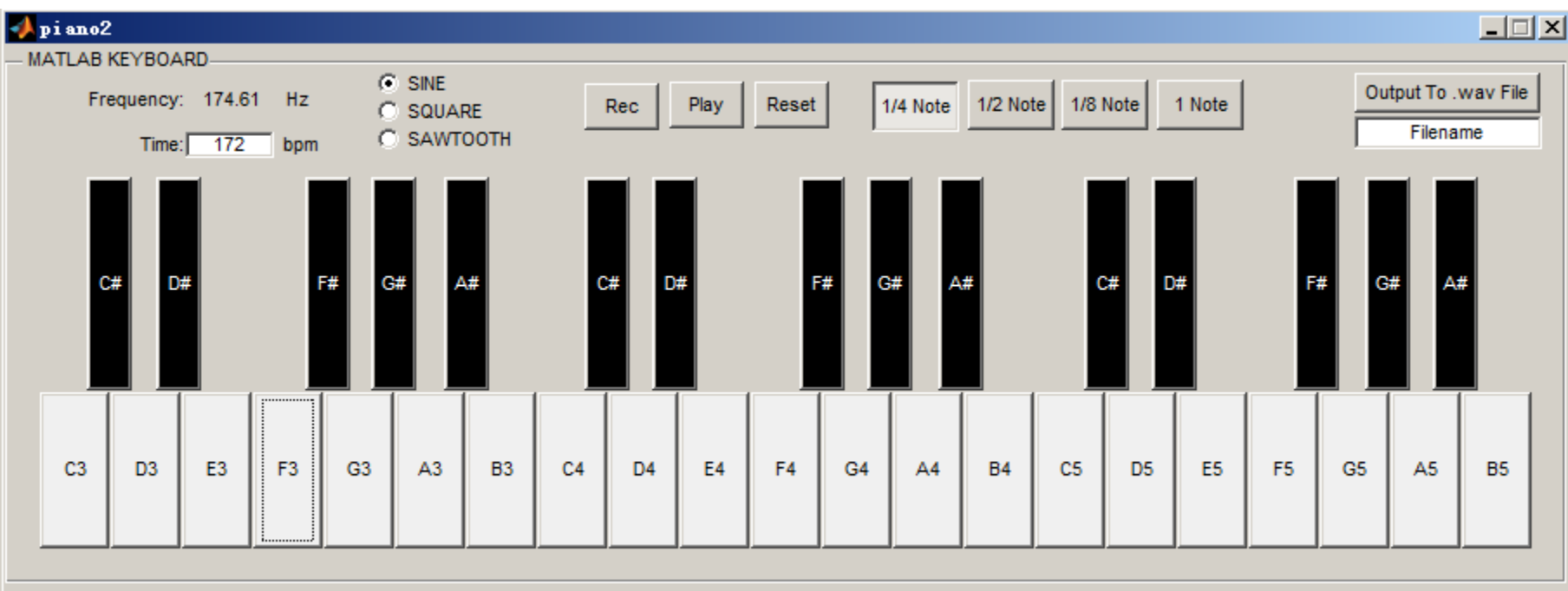
```
handles: 1x1 struct =
    figure1: 186.0018
      text6: 23.0024
      text5: 22.0024
pushbutton1: 21.0024
      t_input: 20.0027
      text4: 19.0106
    f2_input: 18.0106
      text3: 17.0029
    f1_input: 16.0029
      text2: 15.0031
      text1: 197.0016
    time_axes: 192.0016
frequency_axes: 187.0018
      output: 186.0018
```



GUIDE

```
function pushbutton1_Callback(hObject, eventdata, handles)
f1 = str2double(get(handles.f1_input,'String'));
% handles.f1_input: tag属性为f1_input的编辑框的句柄
% get(handles.f1_input,'String')取出该编辑框中
    string%属性的字符串
% str2double(...), 将该字符串转化为double.
f2 = str2double(get(handles.f2_input,'String'));
t = eval(get(handles.t_input,'String'));
% get(handles.t_input,'String')取出的字符串为
0:0.001:0.25, 无法用 str2double函数转换为双精度数
数组, 用eval函数实现
.....
```

GUIDE例5



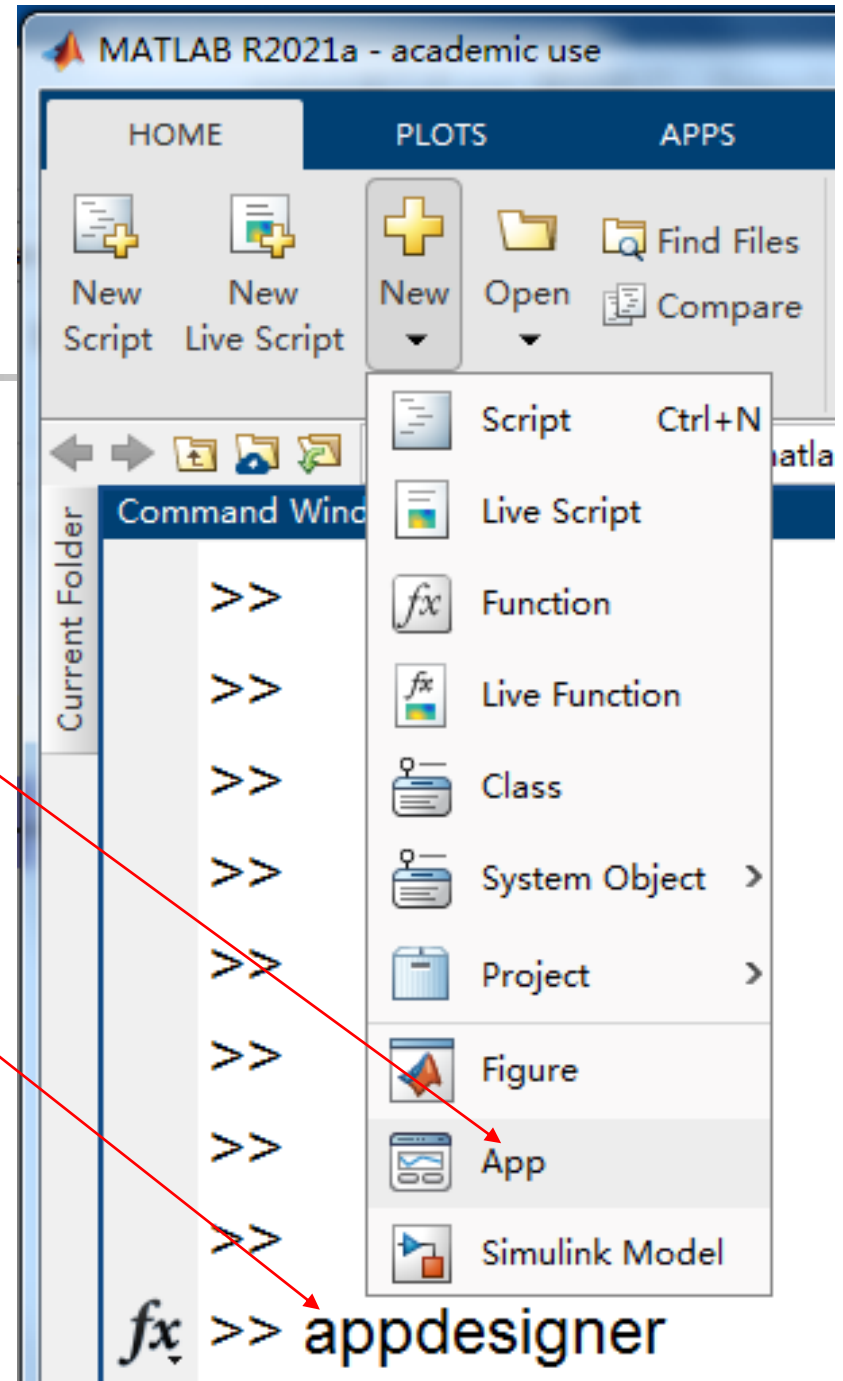


part2

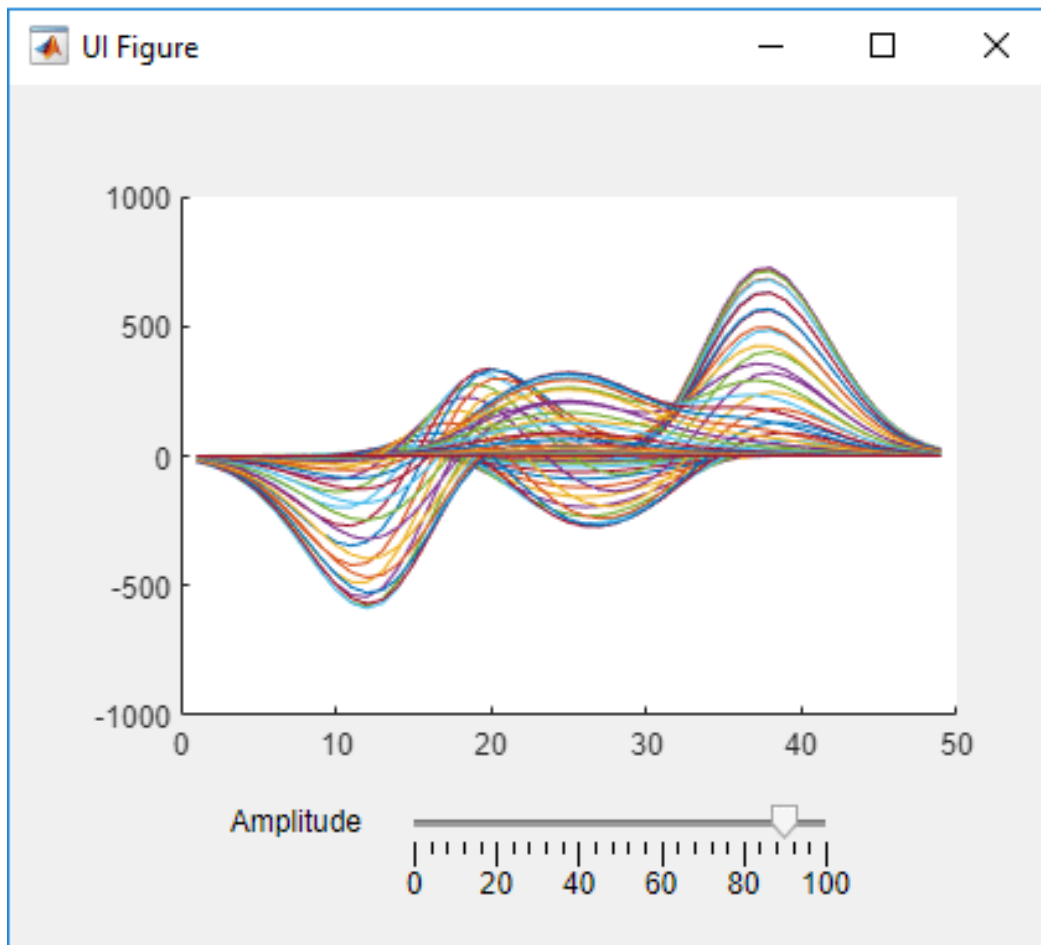
GUI程序设计(三)
——基于AppDesigner

App Designer

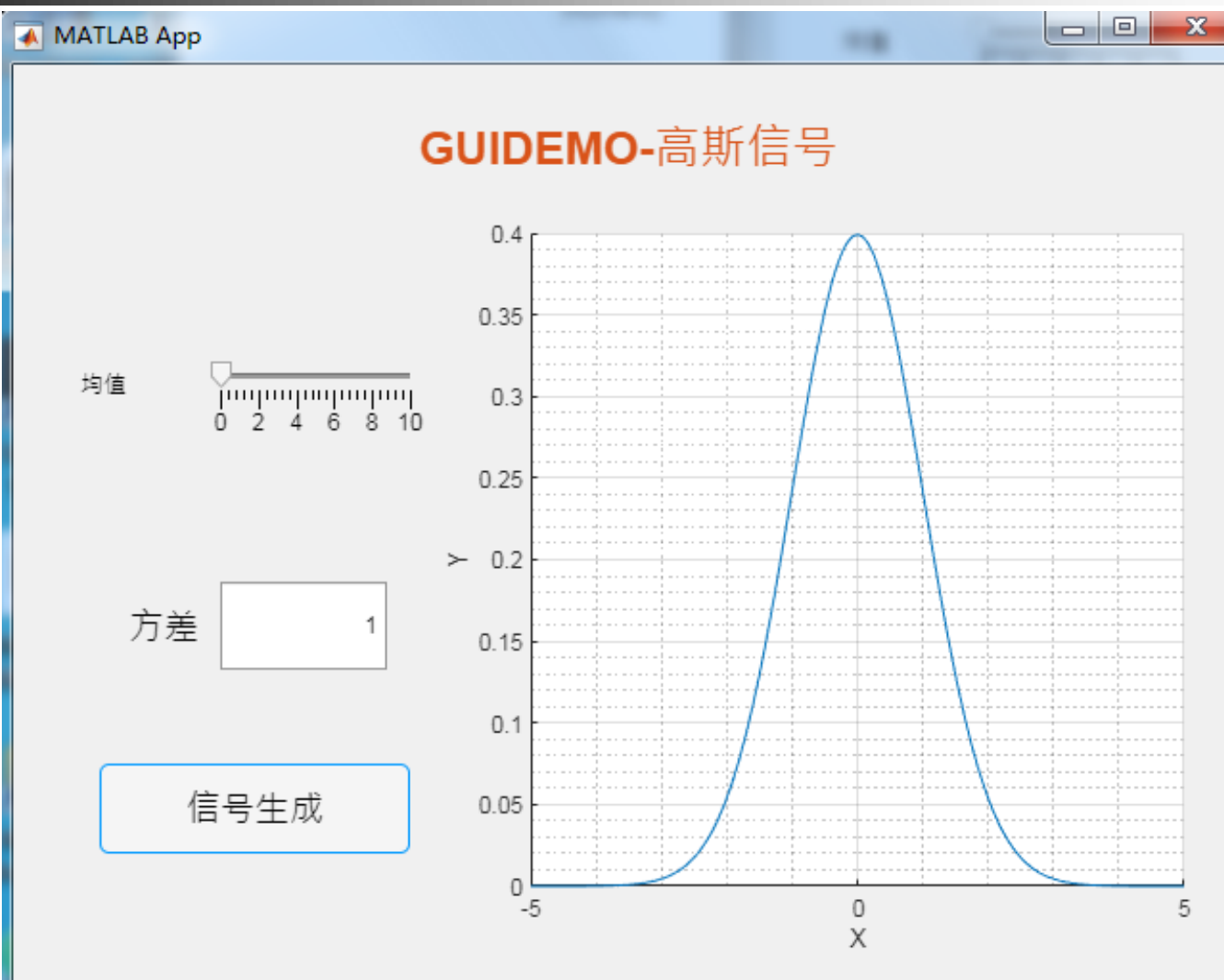
- 在MATLAB中打开AppDesigner程序：
- 在菜单中
- 在命令窗口直接输入



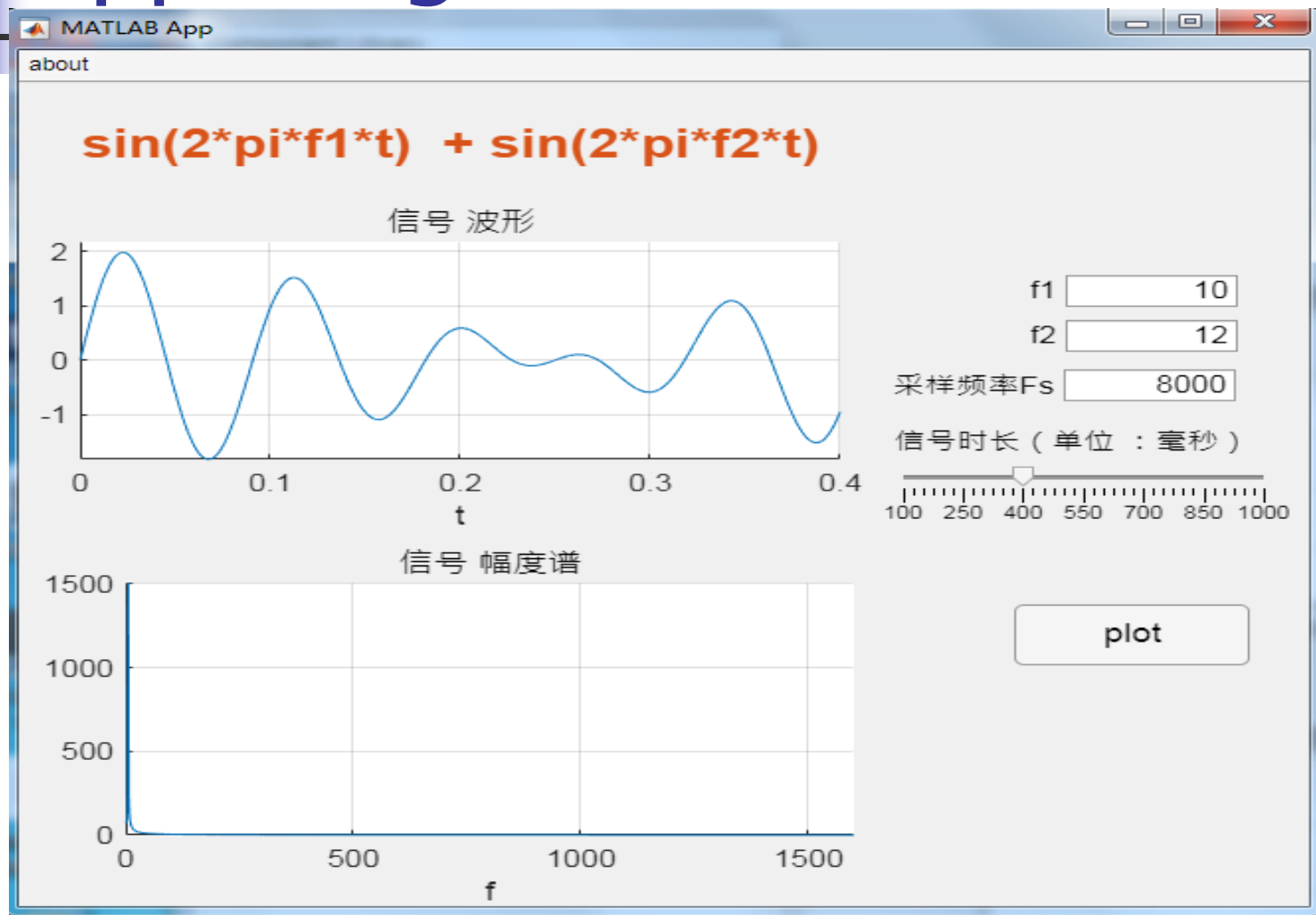
AppDesigner 例1



AppDesigner例2



AppDesigner例3





part3

GUI程序设计(三) ——编程实现



GUI编程实现

■ 例：

%创建一个按钮

```
hp1 = uicontrol('style','pushbutton');
```

%修改按钮属性

```
set(hp1,'position',[100,100,200,200])
```

```
set(hp1,'string','按一下');
```

```
set(hp1,'fontsize',24)
```

%定义回调函数

```
cmd = 'disp(''你按了我一下'')';
```

```
set(hp1,'callback',cmd)
```



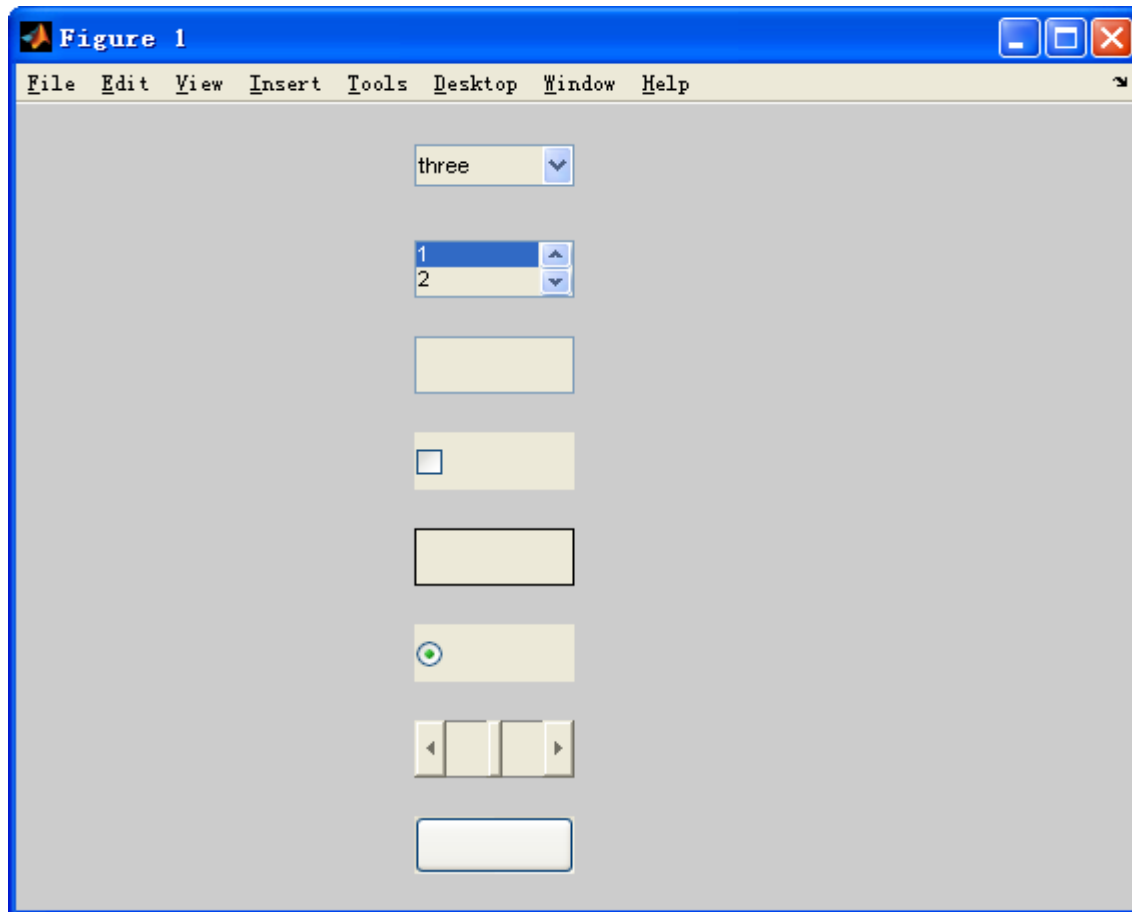
GUI编程实现

- 添加一个关闭按钮

```
hp2 = uicontrol('style','pushbutton');  
set(hp1,'position',[320,100,200,200])  
set(hp1,'string','关闭');  
set(hp1,'fontsize',24)  
set(hp1,'callback','close')
```

GUI编程实现

■ matlab的一些控件





GUI编程实现

```
close all
uicontrol('style','push','position',[200 20 80 30]);
uicontrol('style','slide','position',[200 70 80 30]);
uicontrol('style','radio','position',[200 120 80 30]);
uicontrol('style','frame','position',[200 170 80 30]);
uicontrol('style','check','position',[200 220 80 30]);
uicontrol('style','edit','position',[200 270 80 30]);
uicontrol('style','list','position',[200 320 80
30],'string','1|2|3|4');
uicontrol('style','popup','position',[200 370 80
30],'string','one|two|three');
```

常用控件

'pushbutton'		释放鼠标按键前显示为按下状态的按钮。
'togglebutton'		按钮，它在外观上类似于普通按钮，但在视觉上有状态指示：选中或清除。
'checkbox'		可以单独选择或清除的选项。
'radiobutton'		作为组的一部分的选项，选中它时会清除组中的其他选项。
'edit'		可编辑的文本字段。 about:config 1。

常用控件

'text'

Select an item below:

静态文本字段。通常使用静态文本行为其他用户界面控件添加标签，向用户提供信息，或指示与滑块相关联的值。

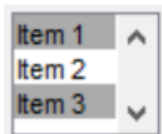
要使静态文本响应鼠标点击，请将 **Enable** 属性设置为 **'Inactive'**，并使用 **ButtonDownFcn** 回调编写响应代码。

'slider'



用户沿水平或垂直滑动条移动的“滑块”按钮。按钮沿条形的位置表示指定范围内的值。

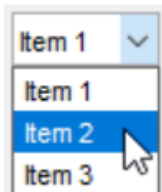
'listbox'



用户可从中选择一项或多项的项列表。与弹出式菜单不同，点击列表框时不会展开。

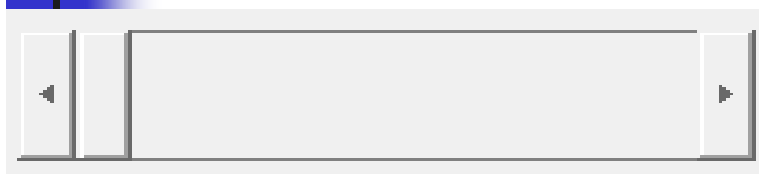
要启用多项选择，请设置 **Max** 和 **Min** 属性以满足 $\text{Max} - \text{Min} > 1$ 。要在可以从一个列表框中选择多项时延迟操作，您可以将**完成**普通按钮与该列表框相关联。然后，使用该按钮的回调来计算列表框 **Value** 属性。

'popupmenu'





弹出式菜单（也称为下拉菜单），展开以显示选项列表。关闭时，弹出式菜单指示出当前选项。如果要提供许多互斥选项，请使用弹出式菜单。

Slider控件

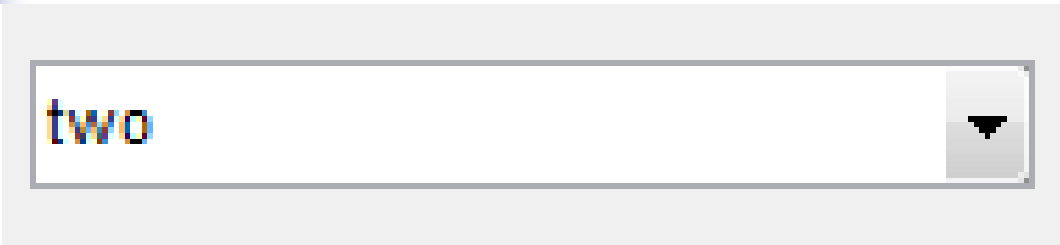


```
hp1 =  
uicontrol('style','slider');  
set(hp1,'position',[100,10  
0,200,50])  
set(hp1,'max',100,'min',0,'  
value',25);
```

■ slider 控件的一些属性:

Max	1.0
Min	0.0
SliderStep	 [1x2 double
Style	slider
Value	 0.45

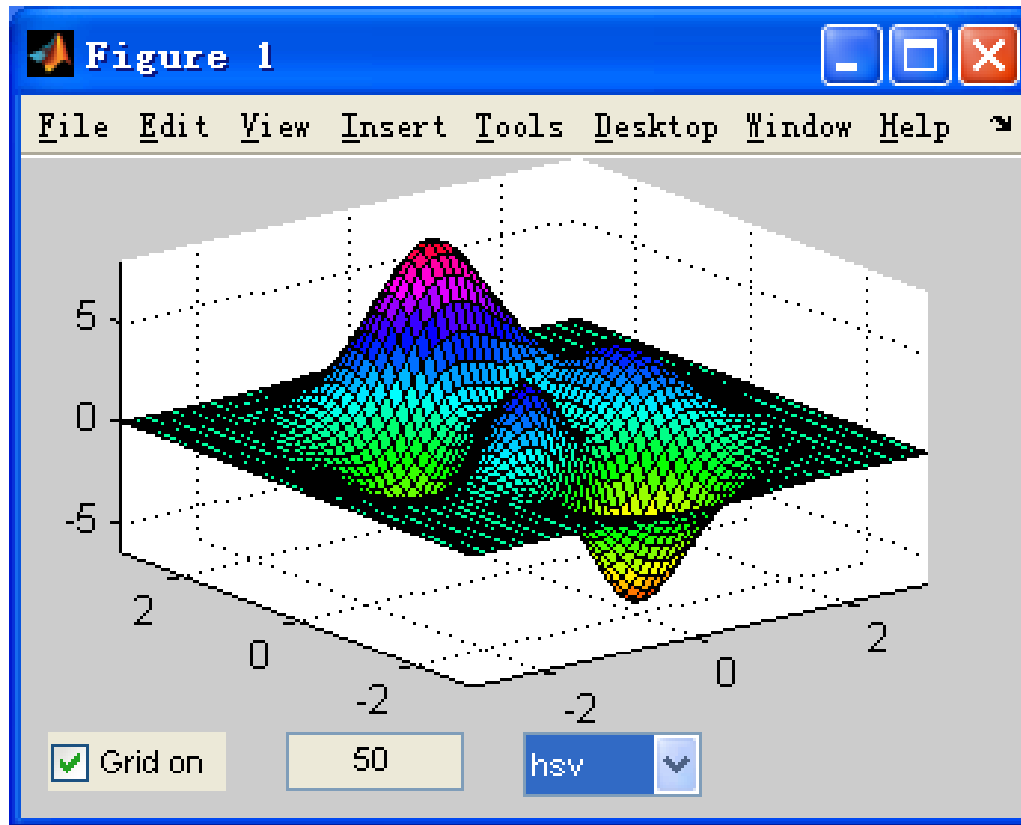
Popupmenu控件



- `h1 = uicontrol('style','popup', ...
 'position',[200 370 180 60],...
 'string','one|two|three');`
`str1 = get(h1,'str')`
`ind = get(h1,'value')`
`str1(ind,:)`

GUI编程实现

- 例: ui01.m





GUI编程实现

ui01.m 代码

```
figure('position', [30 30 300 200]);  
axes('position', [0.1 0.2 0.8 0.8]);  
pointNum = 20;  
[xx, yy, zz] = peaks(pointNum);  
surf(xx, yy, zz);  
axis tight
```

...后面还有



GUI编程实现

■ ui01.m 代码

```
h1 = uicontrol('style', 'checkbox', 'string', 'Grid  
on/off', ...  
    'position', [10, 10, 80, 20], 'value', 1);  
h2 = uicontrol('style', 'edit', 'string',  
int2str(pointNum), ...  
    'position', [100, 10, 60, 20]);  
h3 = uicontrol('style', 'popupmenu', ...  
    'string', 'hsv|hot|cool', ...  
    'position', [180, 10, 60, 20]);  
  
set(h1, 'callback', 'grid');  
set(h2, 'callback', 'cb2');  
set(h3, 'callback', 'cb3');
```

...后面还有



GUI编程实现

■ ui01.m 中文本编辑框控件的回调函数cb2.m

```
pointNum = round(str2num(get(h2, 'string')));  
if pointNum <= 1 | pointNum > 100,  
    pointNum = 10;  
    set(h2, 'string', int2str(pointNum));  
end  
[xx, yy, zz] = peaks(pointNum);  
surf(xx, yy, zz);  
axis tight;  
  
if get(h1, 'value')==1,  
    grid on;  
else  
    grid off;  
end
```



GUI编程实现

- ui01.m中弹出式菜单的回调函数cb3.m

```
switch get(h3, 'value')
    case 1
        colormap(hsv);
    case 2
        colormap(hot);
    case 3
        colormap(cool);
    otherwise
        disp('Unknown option');
end
```




GUI编程实现

- **ui01.m**的潜在问题：
 - 1.需要三个**m**文件，管理不方便
 - 2.使用的变量都在**matlab**基本工作空间中，容易造成变量冲突与覆盖。
- 改进方法：使用**Switch case**结构的程序设计方法
 - 例： **ui02.m**