

第七讲 用户图形界面设计

本章目标

- 理解用户图形界面的用途
- 掌握图形用户界面的设计开发工具GUIDE
- 理解图形对象的常用句柄命令

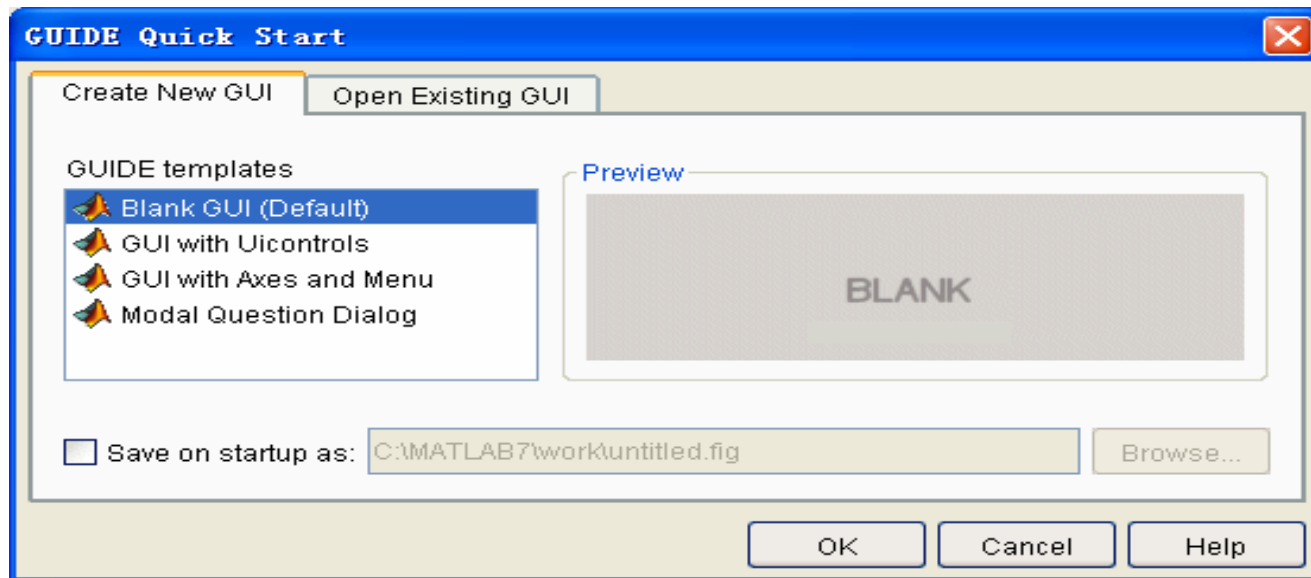
主要内容

- 7.1图形用户界面设计工具**GUIDE**
- 7.2控件
- 7.3 **GUIDE**常用工具
- 7.4使用**GUIDE**的**GUI**设计
- 7.5句柄图形
- 7.6图形对象句柄命令

7.1 图形用户界面设计工具GUIDE

- 可视化图形界面开发环境Guide

Graphical User Interface Development Environment



用户界面开发环境组成

- (1) 菜单栏;
- (2) **Guide**工具栏;
- (3) 图形控件面板;
- (4) 界面编辑客户区。

7.2 控件

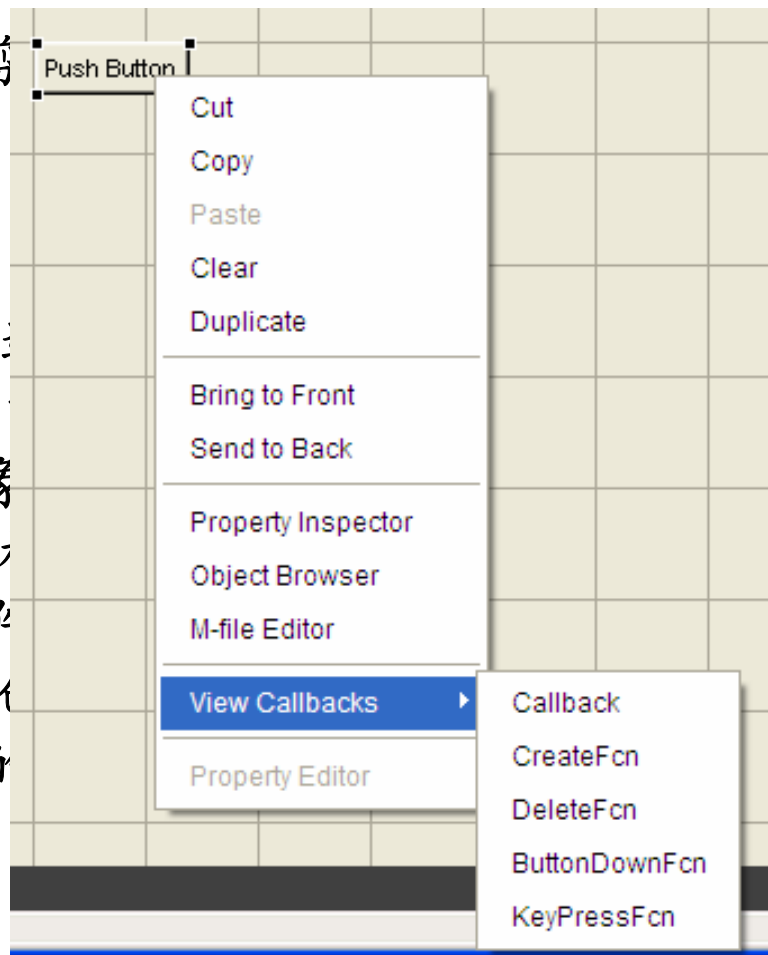
1. 静态文本 (Static Text)
2. 编辑框 (Edit Text) 控件
3. 列表框 (Listbox) 控件
4. 滚动条 (Slider) 控件
5. 按钮 (Push Button) 控件
6. 开关按钮 (Toggle Button) 控件
7. 单选按钮 (Radio Button) 控件
8. 按钮组 (Button Group) 控件
9. 检查框 (Check Box) 控件
10. 列表框 (Listbox) 控件
11. 弹出式菜单 (Popup Menu) 控件
12. 坐标轴 (Axes) 控件
13. 面板 (Panel) 控件

7.3 GUIDE常用工具

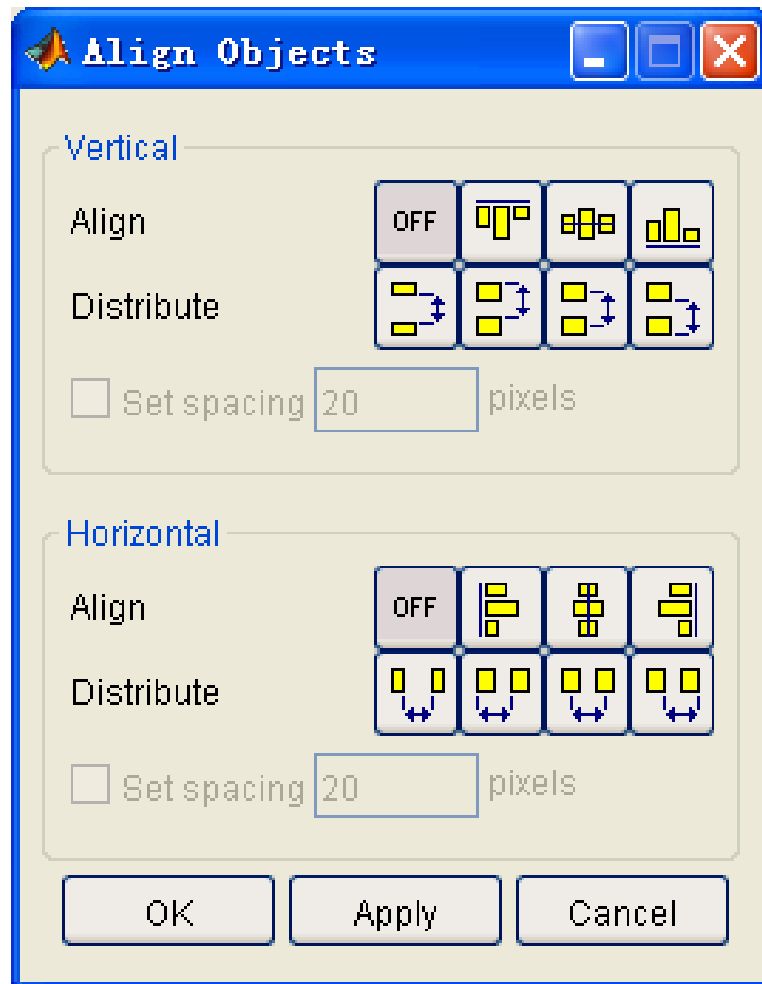
- **Layout Editor:** 布局编辑器;
- **Alignment Tool:** 对齐工具;
- **Property Inspector:** 对象属性观察器;
- **Object Browser:** 对象浏览器;
- **Menu Editor:** 菜单编辑器。

1. 布局编辑器

- (1) **Cut**: 对选中的控件进行剪
- (2) **Copy**: 复制选中的控件。
- (3) **Paste**: 粘贴复制的控件。
- (4) **Delete**: 删除选中的控件。
- (5) **Duplicate**: 对选中的控件
- (6) **Property Inspector**: 对选
- (7) **Object Browser**: 打开对象
- (8) **ButtonDownFcn**: 按下鼠
- (9) **Callback**: 单击鼠标时控件
- (10) **CreateFcn**: 定义控件在
- (11) **DeleteFcn**: 定义在对象的



2. 对齐工具



3. 对象属性观察器

一、控件风格和外观

(1) **BackColor**: 设置控件背景颜色, 使用[R G B]或颜色定义。

(2) **CData**: 在控件上显示的真彩色图像, 使用矩阵表示。

(3) **ForeColor**: 文本颜色。

(4) **String**属性: 控件上的文本, 以及列表框和弹出菜单的选项。

(5) **Visible**: 控件是否可见。

3. 对象属性观察器

二、对象的常规信息

- (1) **Enable**属性: 表示此控件的使能状态, 设置为on”, 表示可选, 为“off”时则表示不可选。
- (2) **Style**: 控件对象类型。
- (3) **Tag**: 控件表示 (用户定义)。
- (4) **TooltipString**属性: 提示信息显示。当鼠标指针位于此控件上时, 显示提示信息。
- (5) **UserData**: 用户指定数据。
- (6) **Position**: 控件对象的尺寸和位置。
- (7) **Units**: 设置控件的位置及大小的单位
- (8) 有关字体的属性, 如 **FontAngle**, **FontName**等。

3. 对象属性观察器

三、控件回调函数的执行

(1) **BusyAction**: 处理回调函数的中断。有两种选项: 即**Cancel**: 取消中断事件, **queue**: 排队(默认设置)。

(2) **ButtonDownFcn**属性: 按钮按下时的处理函数。

(3) **Callback**属性: 是连接程序界面整个程序系统的实质性功能的纽带。该属性值应该为一个可以直接求值的字符串, 在该对象被选中和改变时, 系统将自动地对字符串进行求值。

(4) **CreateFcn**: 在对象产生过程中执行的回调函数。

(5) **DeleteFcn**: 删除对象过程中执行的回调函数。

(6) **Interruptible**属性: 指定当前的回调函数在执行时是否允许中断, 去执行其他的函数。

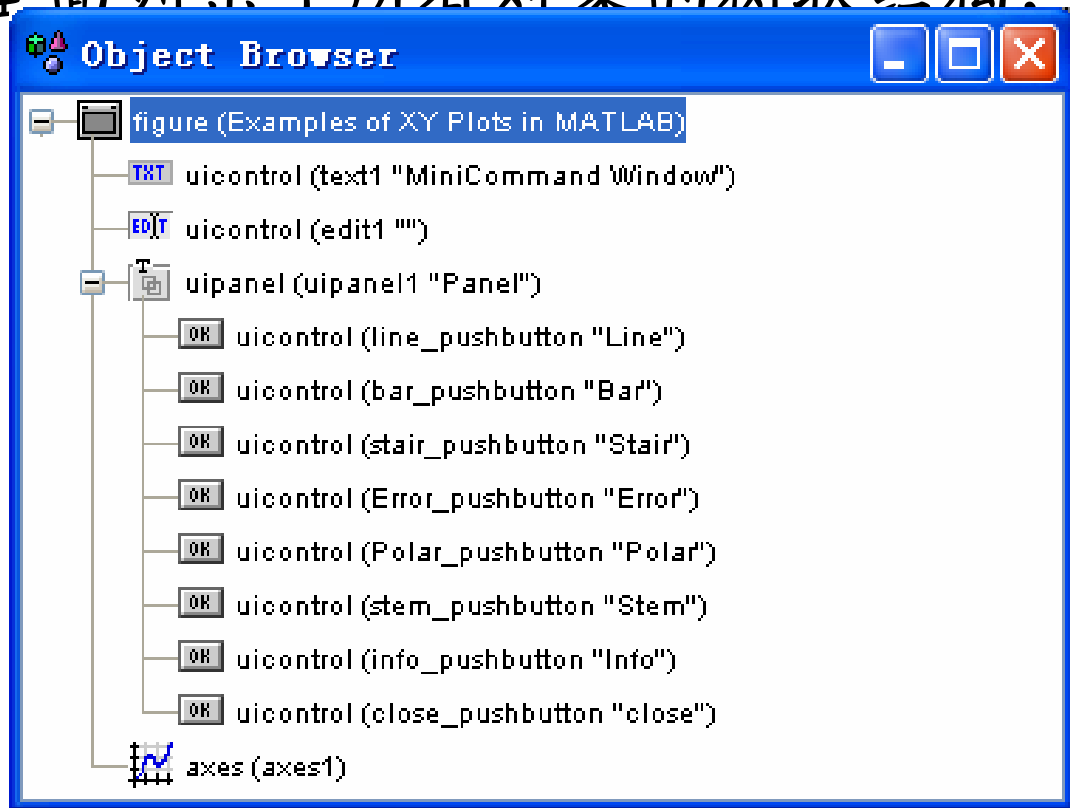
3. 对象属性观察器

四、控件当前状态信息

- (1) **ListboxTop**: 在列表框中显示的最顶层的字符串的索引。
- (2) **Max**: 最大值。
- (3) **Min**: 最小值。
- (4) **Value**: 控件的当前值。

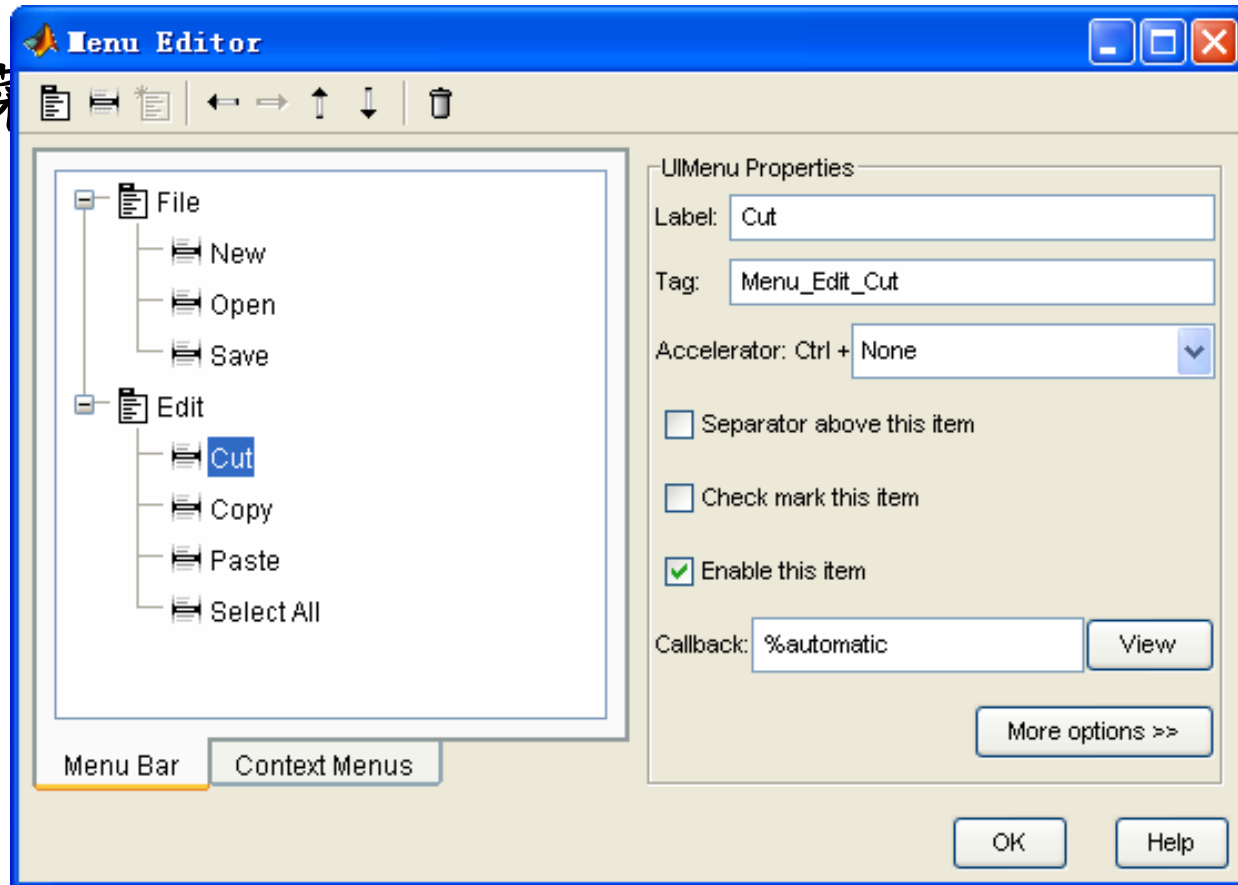
4. 对象浏览器

对象浏览器里面列出了所有对象的树状结构。这些结构就是当所有对象。



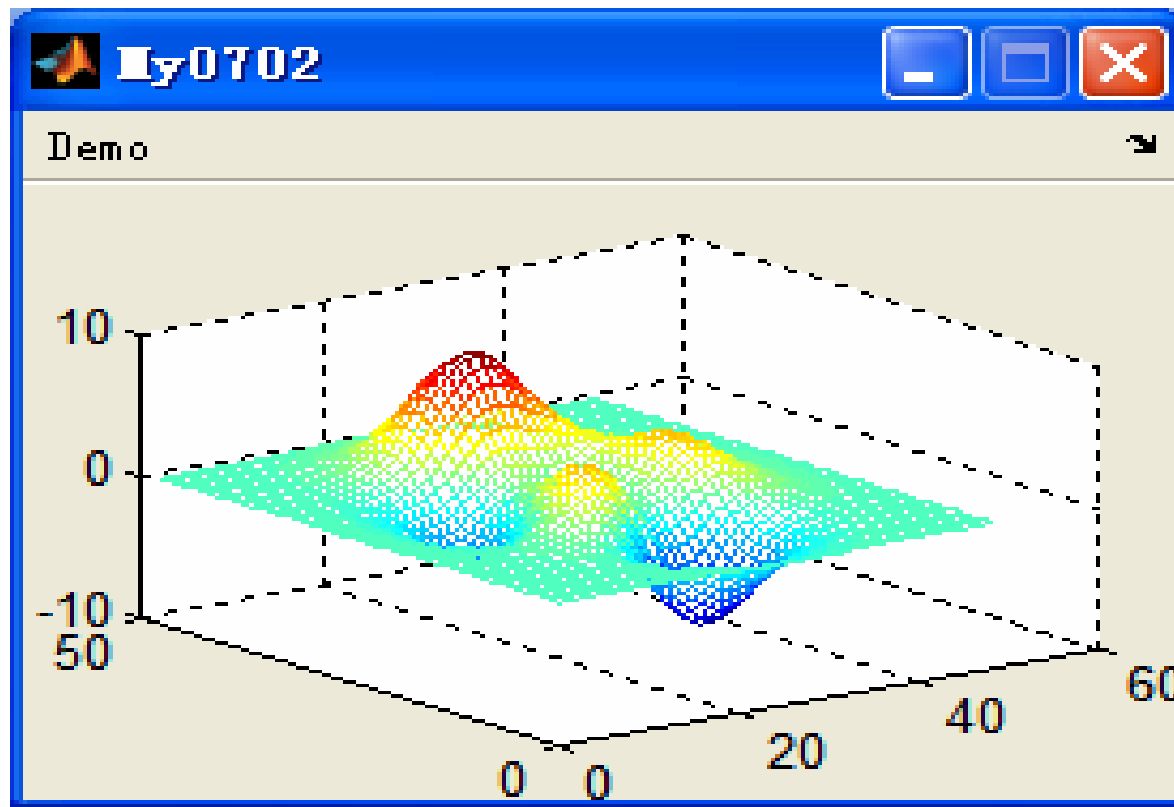
5. 菜单编辑器

菜单



功能:

例7-2 编写一个3D立体图的演示程序



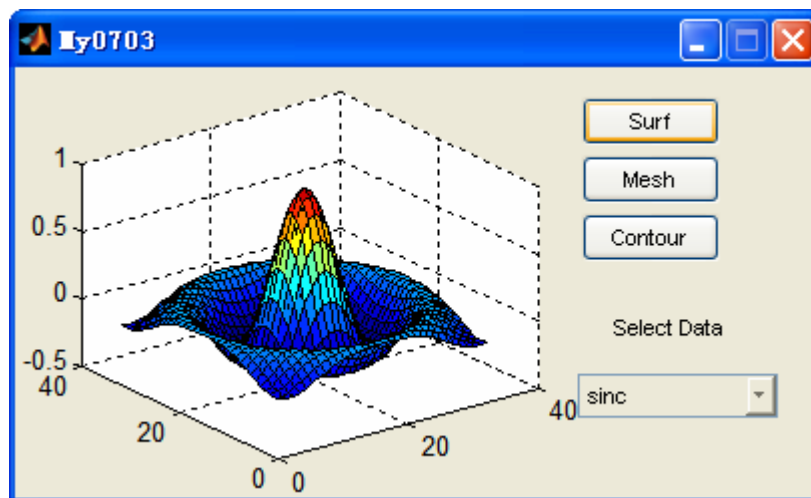
7.4使用GUIDE的GUI设计

- 使用Guide进行GUI的界面设计，就像在纸上绘制界面一样简单。
 - 界面编辑工具 - 设计出界面的大体方案
 - 事件过程编辑器 - 编写各个对象的事件代码

例7-3 建立一个图形界面来显示和处理三维图形

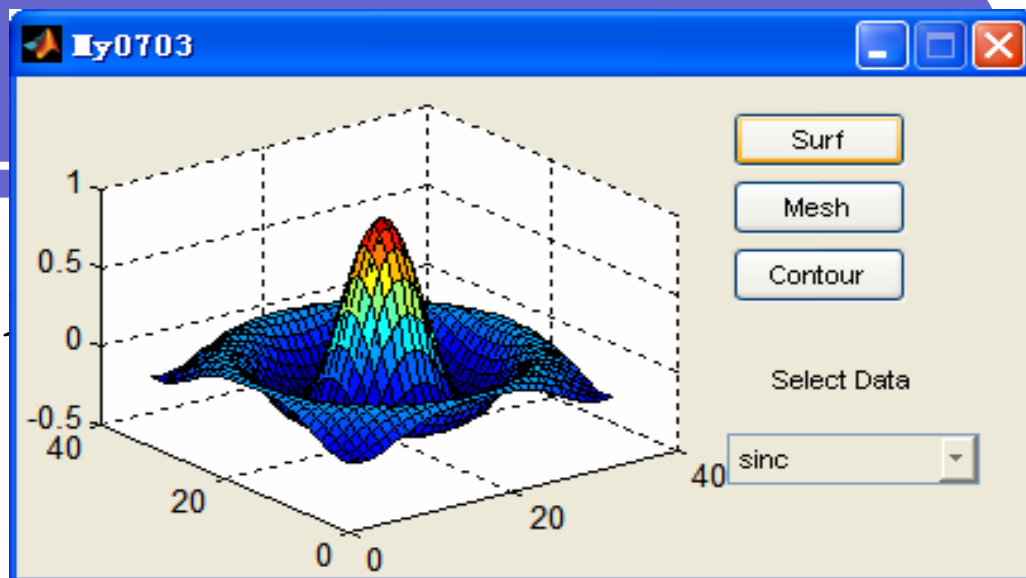
基本功能要求如下。

- (1) 建立一个坐标系，以备以后来绘制三维图形；
- (2) 建立三个按钮，用于启动绘图功能；
- (3) 建立一个组合框（**Popup Menu**），用于显示不同的图形。



解题步骤：

- (1) 打开布局编辑
- (2) 增加组件
- (3) 对齐工具
- (4) 设置每个组件的属性;
- (5) 设置Guide应用程序选项，采用默认属性。
- (6) 激活图形



程序代码：

```
function varargout = My0703(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @My0703_OpeningFcn, ...
    'gui_OutputFcn', @My0703_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
```

程序代码：

```
function My0703_OpeningFcn(hObject, eventdata, handles, varargin)
handles.peak=peaks(35);
handles.membrane=membrane;
[x,y]=meshgrid(-8:0.5:8);
r=sqrt(x.^2+y.^2)+eps;
sinc=sin(r)./r;
handles.sinc=sinc;
handles.current_data=handles.peak;
surf(handles.current_data);
handles.output = hObject;
guidata(hObject, handles);
function varargout = My0703_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function Surf_pushbutton_Callback(hObject, eventdata, handles)
surf(handles.current_data);
function Mesh_pushbutton_Callback(hObject, eventdata, handles)
mesh(handles.current_data);
```

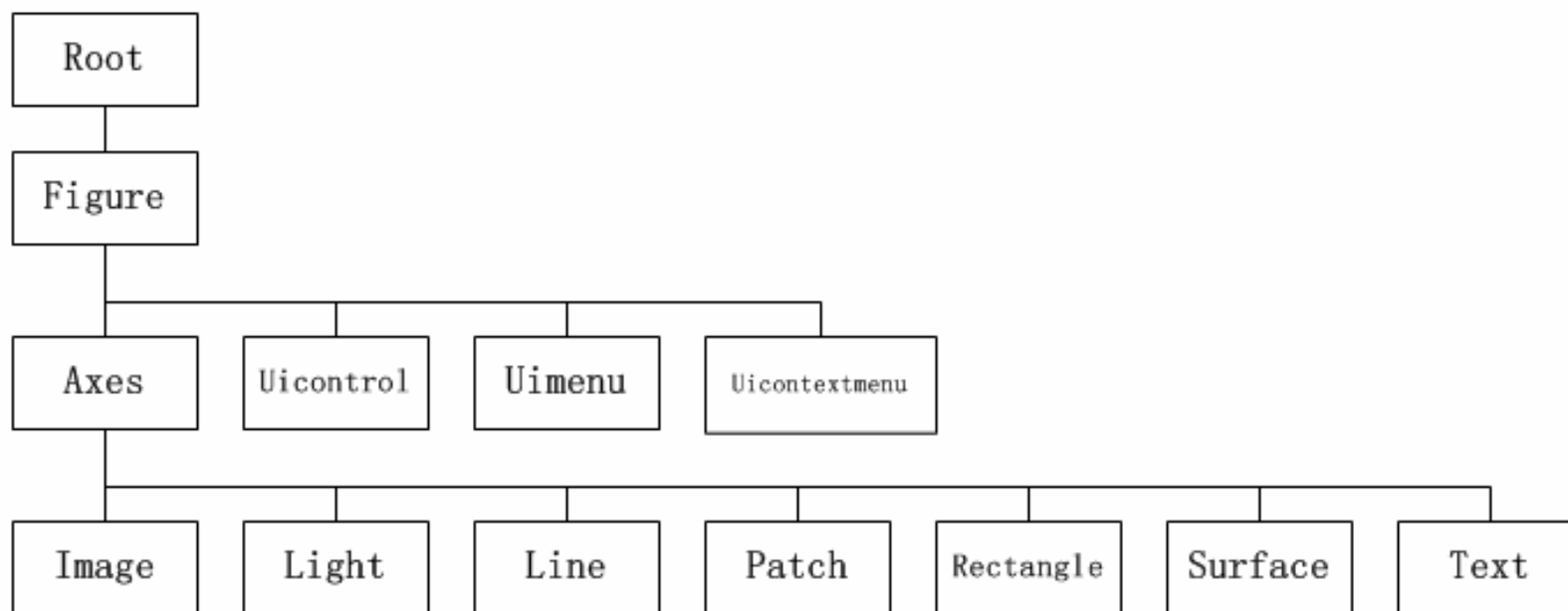
程序代码：

```
function Contour_pushbutton_Callback(hObject, eventdata, handles)  
contour(handles.current_data);  
function data_popup_Callback(hObject, eventdata, handles)  
val=get(hObject,'Value');  
str=get(hObject,'String');  
switch str{val}  
case 'peaks'  
    handles.current_data=handles.peaks;  
case 'membrane'  
    handles.current_data=handles.membrane;  
case 'sinc'  
    handles.current_data=handles.sinc;  
end  
guidata(hObject,handles);
```

7.5 句柄图形

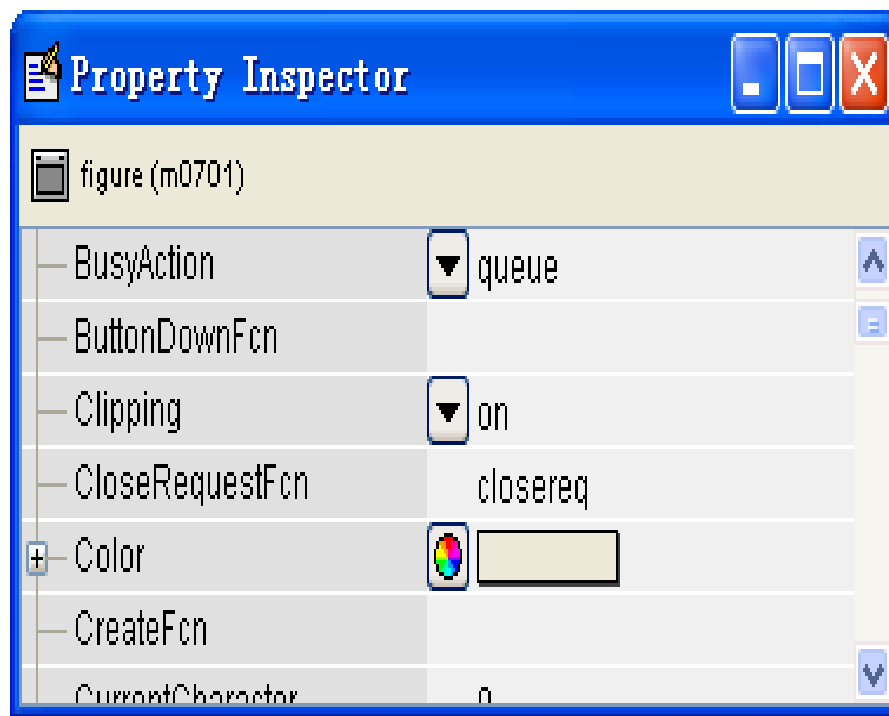
- 句柄图形是对底层图形命令集合的总称，它实际上进行生成图形的工作。

7.5.1 句柄图形的层次关系



7.5.2图形窗口对象的属性

1. Color属性
2. MenuBar属性
3. Name属性
4. NumberTitle属性
5. Units 属性
6. Position属性
7. Resize属性
8. Visible属性
9. NextPlot属性
10. 打印纸属性
11. 各种回调函数



7.6图形对象句柄命令

- 句柄系统可以从底层(low level)创建和操作线、面、文字、图像等基本图形对象指令。

7.6.1 建立图形对象的函数

1. **figure**函数：创建一个新的图形对象。
2. **newplot**函数：做好开始画新图形对象的准备。
3. **axes**函数：创建坐标轴图形对象。
4. **line**函数：画线。
5. **patch**函数：填充多边形。
6. **surface**函数：绘制三维曲面。
7. **image**函数：显示图片对象。
8. **uicontrol**函数：生成用户控制图形对象。
9. **uimenu**函数：生成图形窗口的菜单中层次菜单与下一级子菜单。

例7-4 显示曲线图

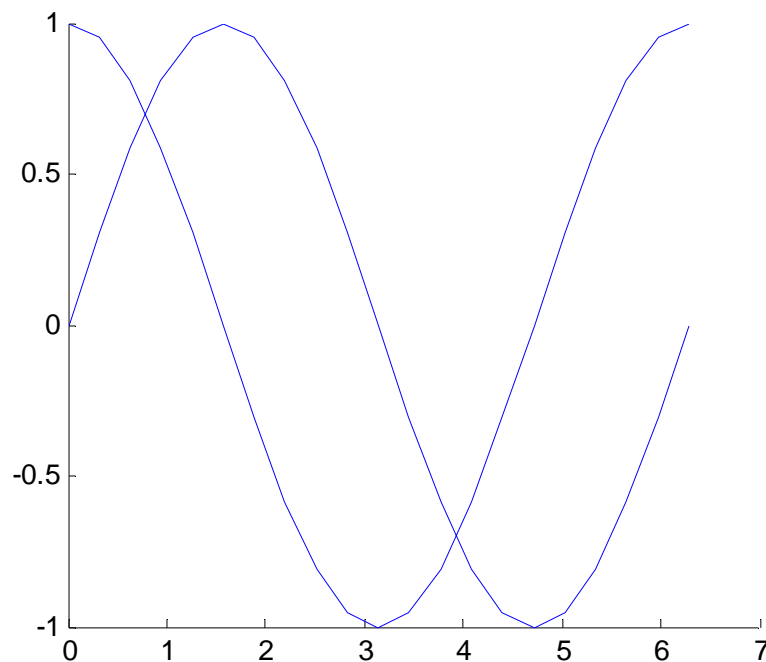
程序:

```
h=figure();
```

```
t=0:pi/10:2*pi;
```

```
line(t, sin(t));
```

```
line(t, cos(t));
```

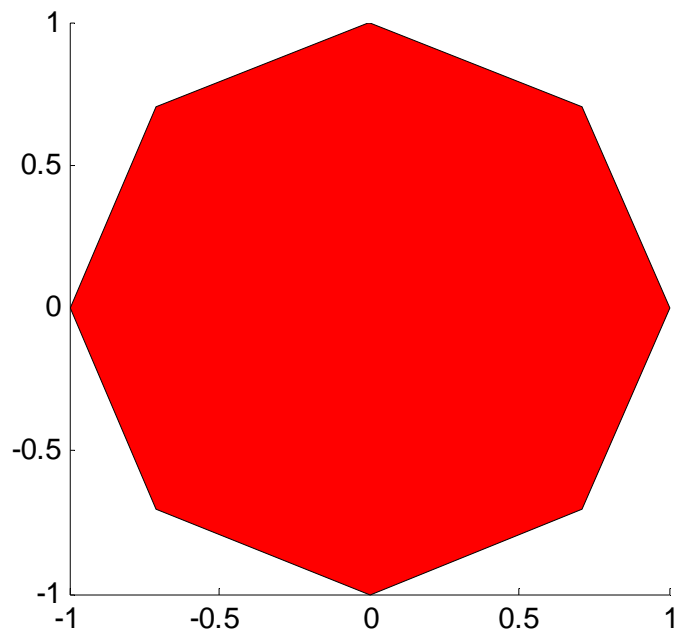


例7-5显示多边形。

程序:

```
x=0:pi/4:2*pi;
```

```
patch(sin(x),cos(x),'r')
```



7.6.2图形对象句柄的获得和删除

- **gcf**函数：获得当前图形窗口的句柄
- **gca**函数：获得当前坐标轴的句柄
- **gco**函数：获得当前对象的句柄
- **gcbo**函数：获得当前正在执行调用的对象的句柄
- **gcbf**函数：获取包括正在执行调用的对象的图形句柄
- **delete**函数：删除句柄所对应的图形对象
- **findobj**函数：查找具有某种属性的图形对象

7.6.3对象属性的获取和设置

- (1) **get**函数返回某些对象属性的当前值。例如:
 - `p = get(hf_1,'Position');`
- (2) 函数**set**改变句柄图形对象属性，例如:
 - `set(hf_1,'Position',vect);`

扩展阅读

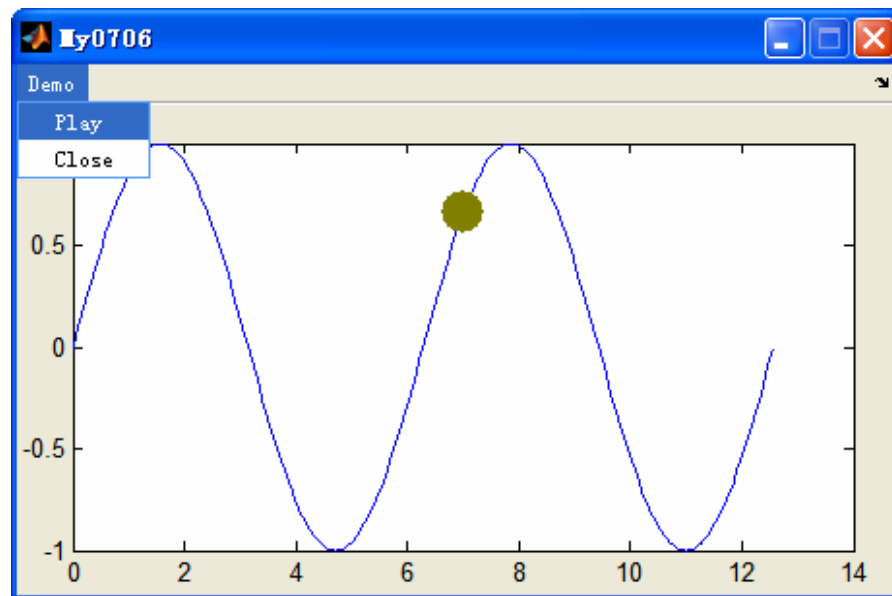
● 7.7公用对话框文件

- 文件选择对话框
- 颜色选择对话框
- 字体选择对话框
- 页面设置对话框
- 警告与错误信息对话框
- 简单帮助信息对话框

上机指导

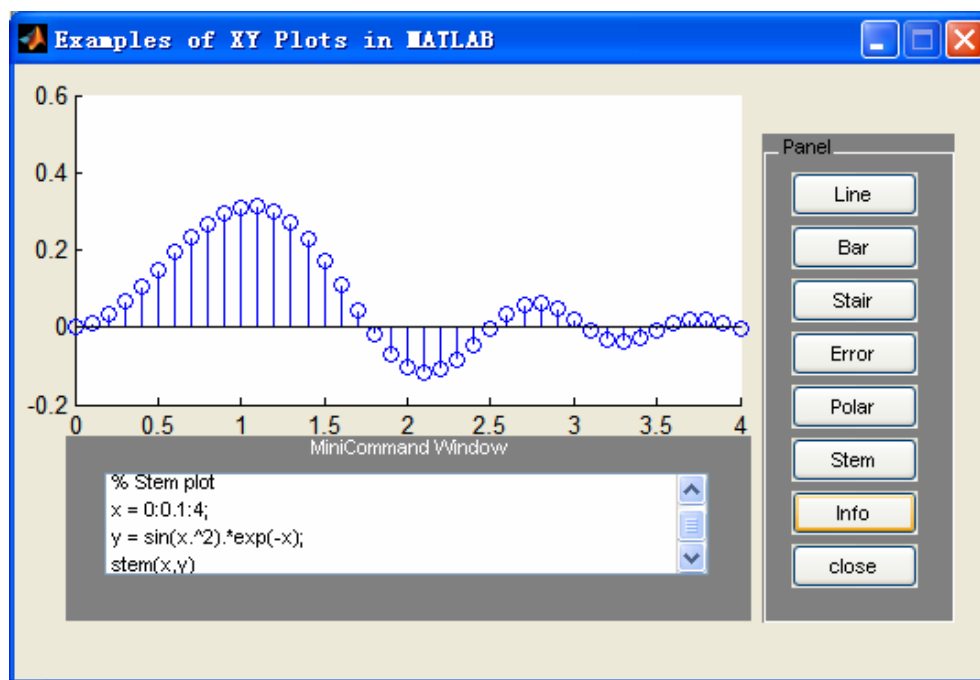
7.8 动画制作

- 例：编写一个演示程序，用于显示小球沿正弦曲线运动的运动规律。



应用举例

利用界面设计工具guide，重新编写MATLAB中平面演示程序graf2d.m。



解题步骤：

- （1）利用界面编辑器，设计窗口初始位置和大小，添加所需控件
- （2）新建图形对象，对齐所有控件
- （3）设置新建对象的属性
- （4）在产生的M文件加入相应代码

程序代码 — 1

```
function line_pushbutton_Callback(hObject, eventdata, handles)  
cmdStr=str2mat( ...  
    ' % Line plot of a chirp', ...  
    ' x=0:0.05:5;', ...  
    ' y=sin(x.^2);', ...  
    ' plot(x,y);' ...  
);  
set(handles.edit1,'String',cmdStr);  
evalmcw(handles.edit1);  
function bar_pushbutton_Callback(hObject, eventdata, handles)  
cmdStr=str2mat( ...  
    ' % Bar plot of a bell shaped curve', ...  
    ' x = -2.9:0.2:2.9;', ...  
    ' bar(x,exp(-x.*x));' ...  
);  
set(handles.edit1,'String',cmdStr);  
evalmcw(handles.edit1);
```

程序代码 – 2

```
function stair_pushbutton_Callback(hObject, eventdata, handles)
cmdStr=str2mat( ...
    ' % Stairstep plot of a sine wave', ...
    ' x=0:0.25:10;', ...
    ' stairs(x,sin(x));' ...
);
set(handles.edit1,'String',cmdStr);
evalmcw(handles.edit1);
function Error_pushbutton_Callback(hObject, eventdata, handles)
cmdStr=str2mat( ...
    ' % Errorbar plot', ...
    ' x=-2:0.1:2;', ...
    ' y=erf(x);', ...
    ' e = rand(size(x))/10;', ...
    ' errorbar(x,y,e);' ...
);
set(handles.edit1,'String',cmdStr);
evalmcw(handles.edit1);
```

程序代码 — 3

```
function Polar_pushbutton_Callback(hObject, eventdata, handles)
cmdStr=str2mat( ...
    ' % Polar plot', ...
    ' t=0:.01:2*pi;', ...
    ' polar(t,abs(sin(2*t).*cos(2*t))); ' ...
);
set(handles.edit1,'String',cmdStr);
evalmcw(handles.edit1);
function stem_pushbutton_Callback(hObject, eventdata, handles)
cmdStr=str2mat( ...
    ' % Stem plot', ...
    ' x = 0:0.1:4;', ...
    ' y = sin(x.^2).*exp(-x);', ...
    ' stem(x,y)' ...
);
set(handles.edit1,'String',cmdStr);
evalmcw(handles.edit1);
```

程序代码 – 4

```
function info_pushbutton_Callback(hObject, eventdata, handles)  
helpwin  
function close_pushbutton_Callback(hObject, eventdata, handles)  
close(gcf);
```