Updating table via popupmenu matlab gui

I'm writing a matlab GUI which contains popupmenu and a table. My goal is to choose option from the popupmenu and it will fill the table according to it.

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

% Hint: popupmenu 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

handles.Item_list_value = get(hObject,'Value');
```

After this line im receiving the option number that I chose from the popupmenu and I would like to take the same column number from a matrix and insert it into the table.

something like:

```
set(handles.Table,'data',)...
```

(I want to insert into the 1st column the 1st column from matrix Database)

However at this point nothing works. The tag of the table is Table.

matlab user-interface

edited Nov 17 '16 at 20:58 — Brian Tompsett - 汤莱恩 — 3,953 13 34 80

asked Jan 1 '16 at 10:35 — Ben — 167 2 14

1 Answer

In the code you've posted, the instruciton to et the index of the selected popupmenu item (`handles.Item_list_value = get(hObject,'Value');`) is written in the popupmenu `CreateFcn` , unless you did not post the right code, this implies that you can get the index of the selected item only once, during the creation of the popupmenu.

To insert / update the content of a `uitable` you should code the following procedure in the `popupmenu callback` :

- Get the index of the selected popupmenu item with the `get` function
- Get the table data with the `get` function (the data are returned as (n x m) array)
- Insert the input matrix column selected with popupmenu into the proper table column: the input matrix should be "visible"in the `popupmenu callback` . In your question is not specified how you manage the input matrix, one posibility could be to store it (after, for example having read it form an input file) into the GUI data by using the `guidata` function
- Updata the table data (with the `set` function)

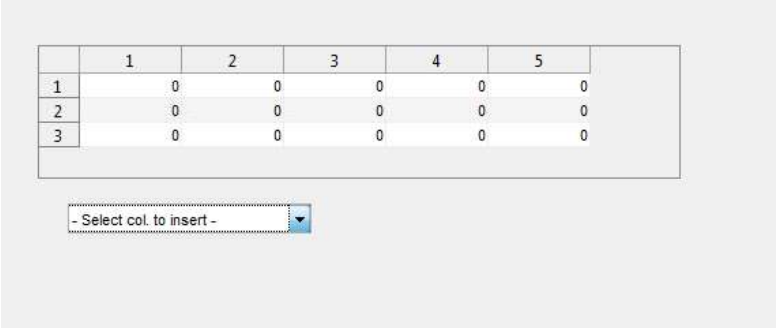
I've built a small GUI (updating_table_via_popup) to test this approach: - popupmenu tag: popupmenu1 - table tag: uitable1 - the input matrix is defined as a set of random numebers in the GUI `OpeningFcn` - the strings to be displayed in the popupmenu are built in the `OpeningFcn` - the input matrix is stored in the in the handles struct of the GUI data

This is the code of the GUI, ref. to the `updating_table_via_popup_OpeningFcn` and `popupmenu1_Callback` function for the implementation of the proposed solution.

```
function varargout = updating_table_via_popup(varargin)
% UPDATING_TABLE_VIA_POPUP MATLAB code for updating_table_via_popup.fig
%      UPDATING_TABLE_VIA_POPUP, by itself, creates a new UPDATING_TABLE_VIA_POPUP or
raises the existing
%      singleton*.
%
%      H = UPDATING_TABLE_VIA_POPUP returns the handle to a new UPDATING_TABLE_VIA_POPUP
or the handle to
%      the existing singleton*.
```

```
%
%      UPDATING_TABLE_VIA_POPUP('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in UPDATING_TABLE_VIA_POPUP.M with the given input
arguments.
%
%      UPDATING_TABLE_VIA_POPUP('Property','Value',...) creates a new
UPDATING_TABLE_VIA_POPUP or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before updating_table_via_popup_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to updating_table_via_popup_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

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

% Last Modified by GUIDE v2.5 01-Jan-2016 19:55:39

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State =                                    ..

                                                  ia_popup_OpeningFcn, ...
                                                  popup_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


% --- Executes just before updating_table_via_popup is made visible.
function updating_table_via_popup_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 updating_table_via_popup (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

% Define the input matrix
n_row=3;
n_col=5;
data_base=reshape([1:n_row*n_col]',n_col,n_row)';
% Get the GUI data
my_gui_data=guidata(gcf);
% Store the input matrix into the GUI data
my_gui_data.data_base=data_base;
% Update GUI data
guidata(gcf,my_gui_data);
% Initialize table to "0"
set(handles.uitable1,'data',zeros(n_row,n_col))
% Build the popupmenu strings
str{1}='- Select col. to insert -'
for i=2:n_col+1
   str{i}=['Insert Col. #' num2str(i-1)]
end
% Assign the strings to the popupmenu
set(handles.popupmenu1,'string',str)


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


% --- Outputs from this function are returned to the command line.
function varargout = updating_table_via_popup_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;


% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1

% Get the GUI data
my_gui_data=guidata(gcf);
% Get the index of the selected popupmenu item (-1 to account for the first
% string)
sel_idx=get(handles.popupmenu1,'value')-1;
% Get the table data
table_data=get(handles.uitable1,'data');
% Insert the input matrix column selected with popupmenu into the proper
% table column
table_data(:,sel_idx)=my_gui_data.data_base(:,sel_idx)
% Updata the table data
set(handles.uitable1,'data',table_data);


% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu 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
```



Hope this helps.

answered Jan 1 '16 at 20:56

il_raffa
4,147 10 18 29