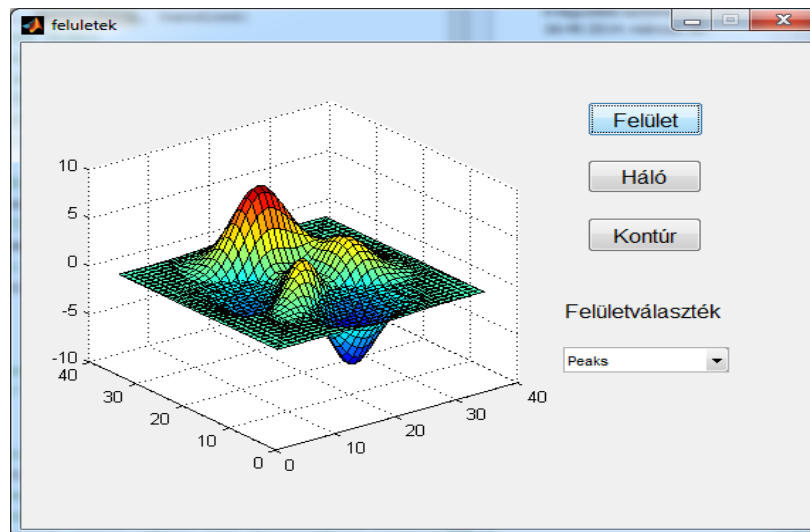


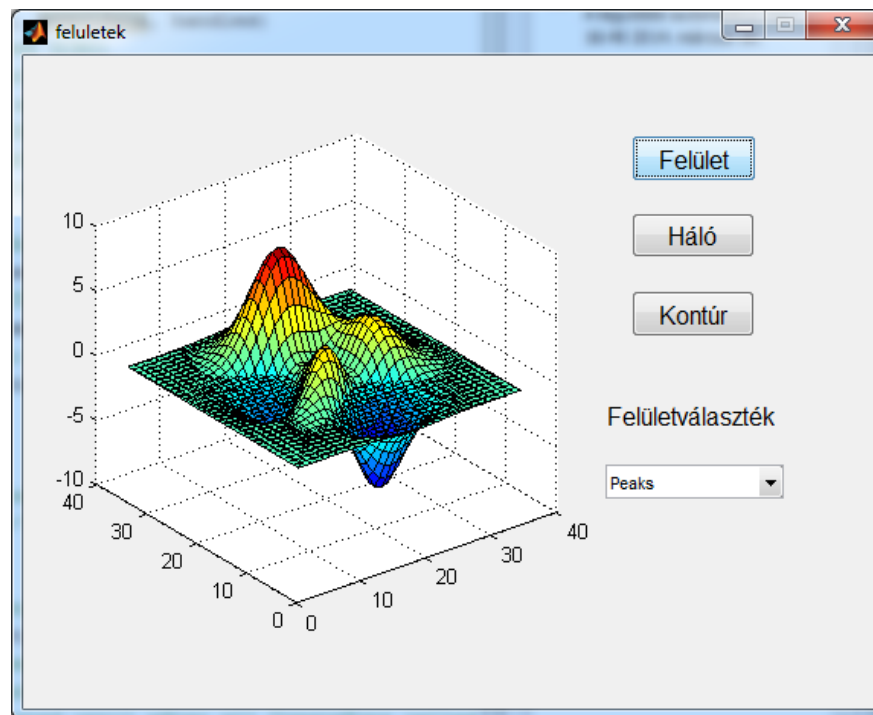
- **MATLAB alapismeretek X.**

- *Egy összetettebb példa grafikus felhasználói felület (GUI)*
létrehozására

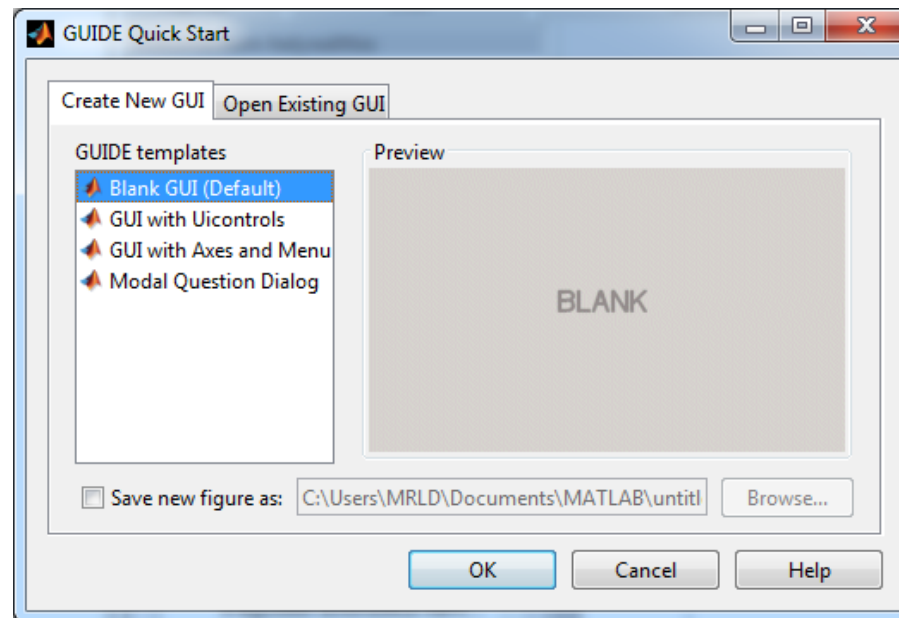


- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

- Készítsük el az alábbi ábrán látható alkalmazást, amelyen legördülő menüből választhatunk háromféle felület közül és bármelyik kirajzoltatható felületként, felületi hálóval, vagy alaplapra vetített szintvonalalaival.

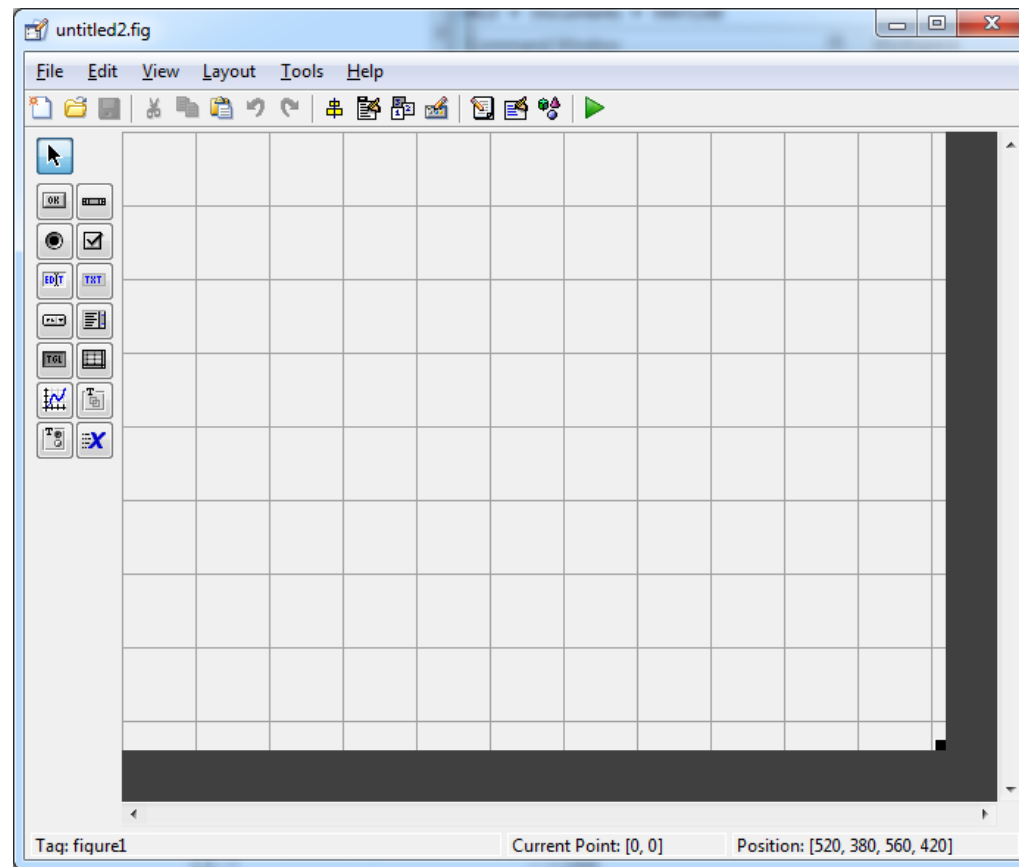


- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Nyissunk meg egy új GUI-t az elrendezés szerkesztőben: írjuk be a parancsablakban: **guide**



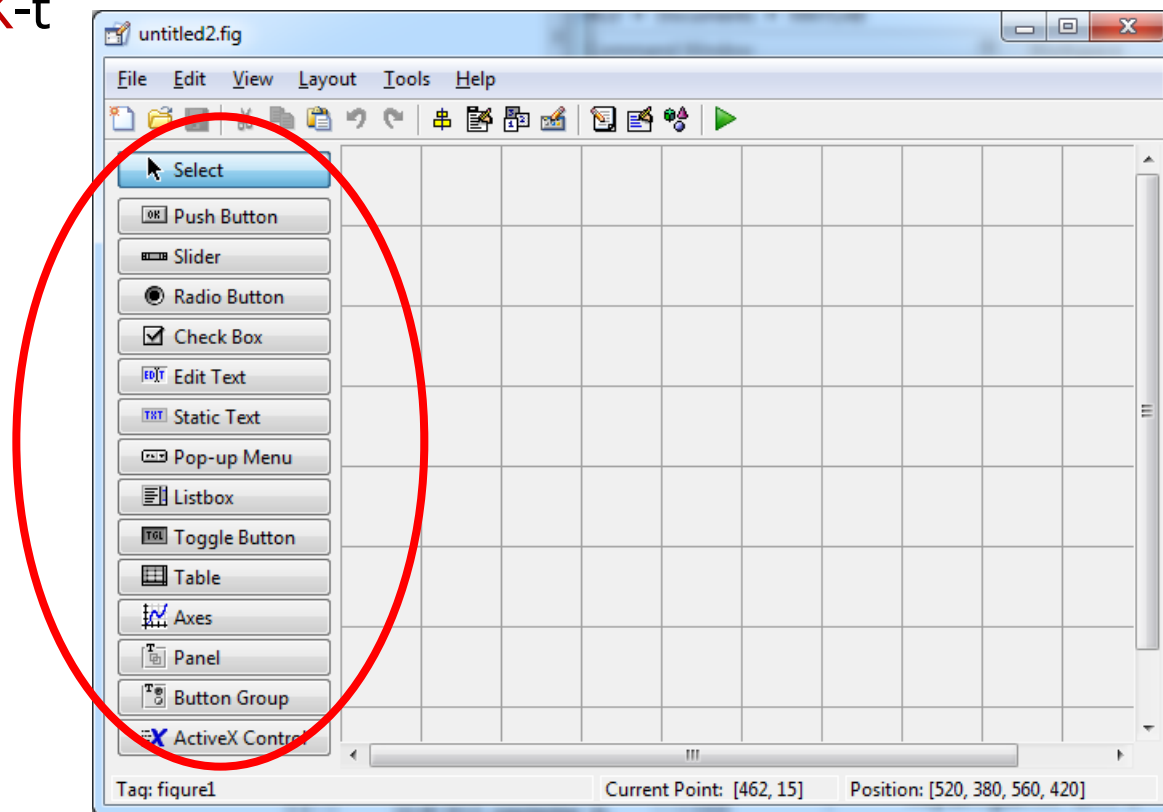
- Válasszuk a Blank, üres GUI-t

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Az üres elrendezés-szerkesztő:

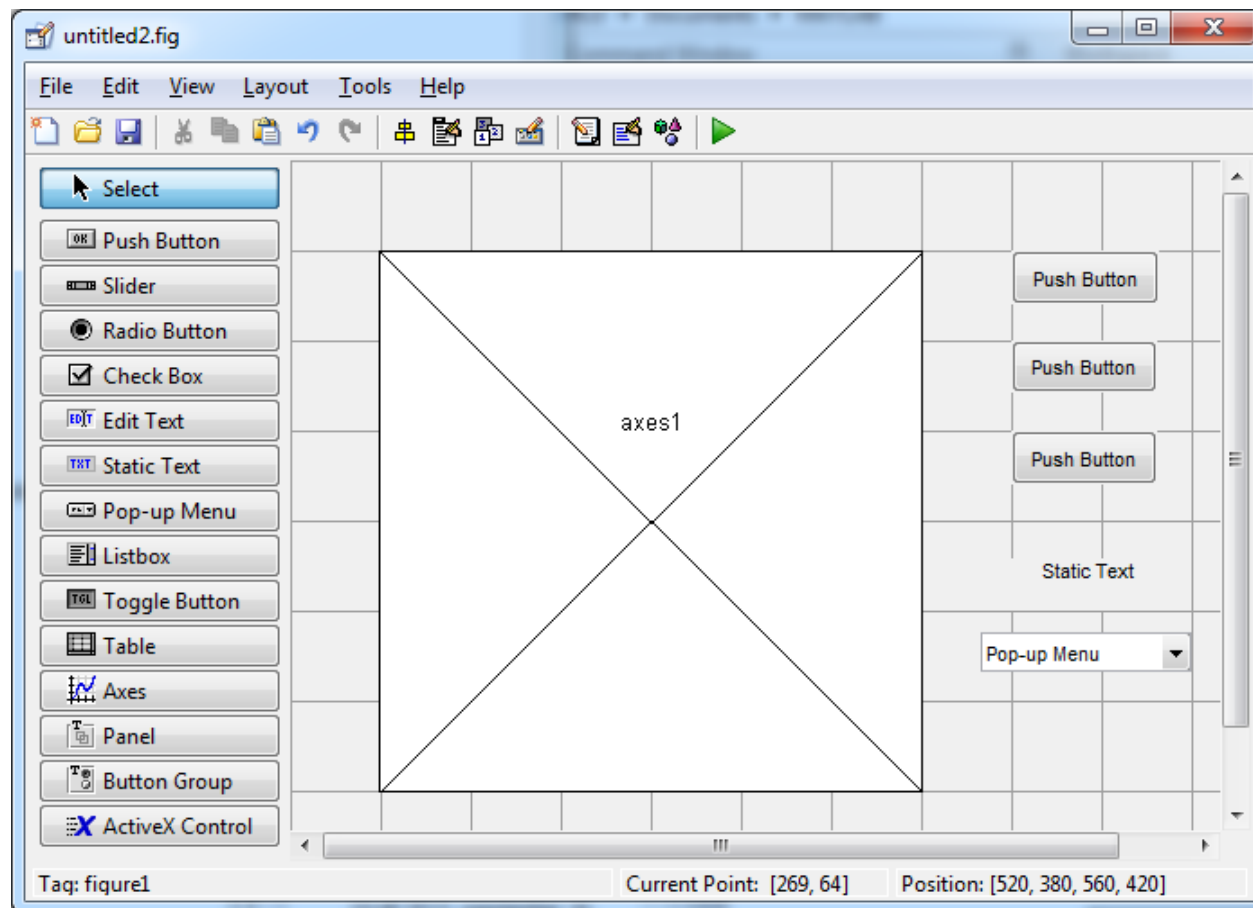


- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

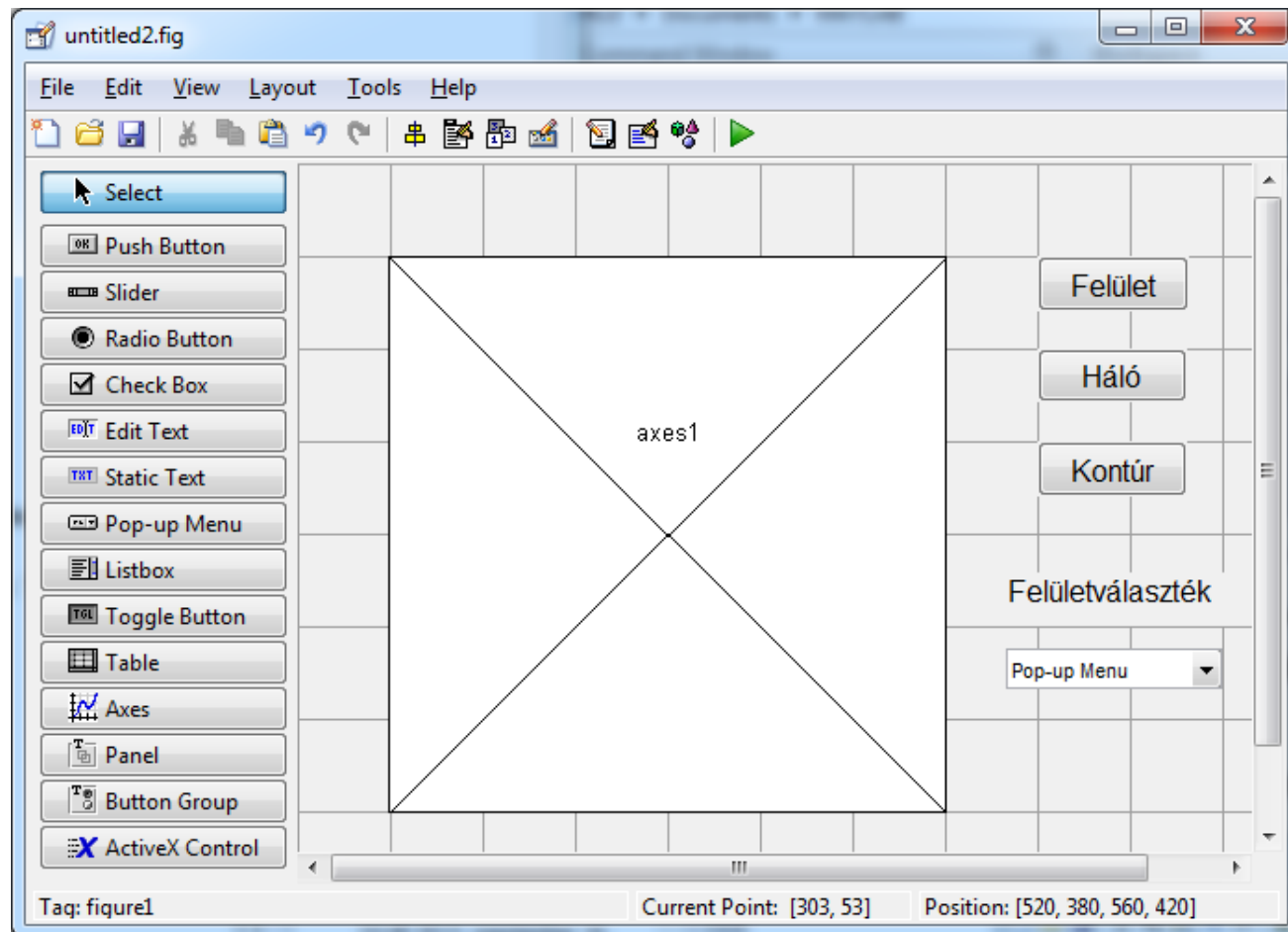
- Jelenítsük meg a komponenspalettán lévő komponensek nevét:
 - **File/Preferences/GUIDE**
 - Válasszuk: **Show names in component palette**
 - Nyomjunk **OK**-t
- Az eredmény:



- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Helyezzük el a komponenseket a szerkesztőfelületen, pozicionáljuk, méretezzük:

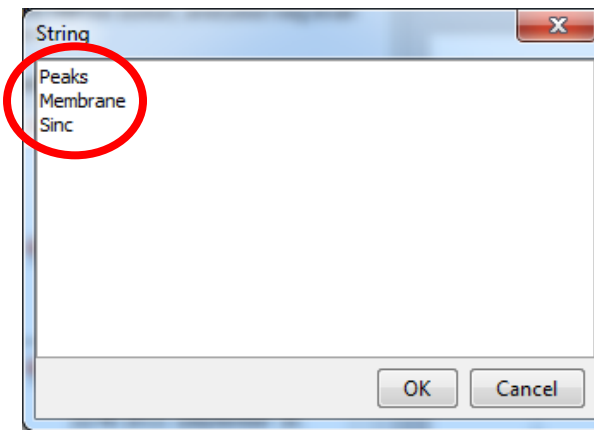
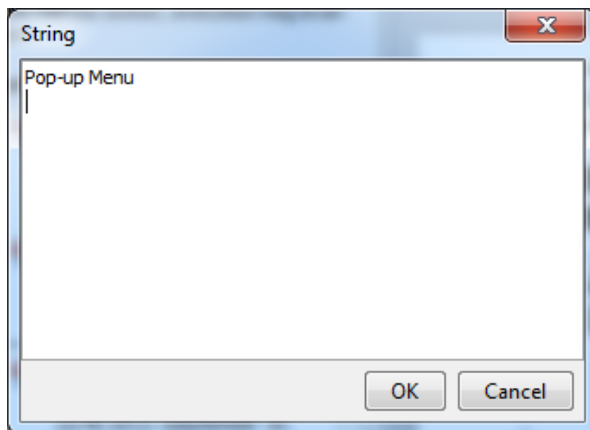
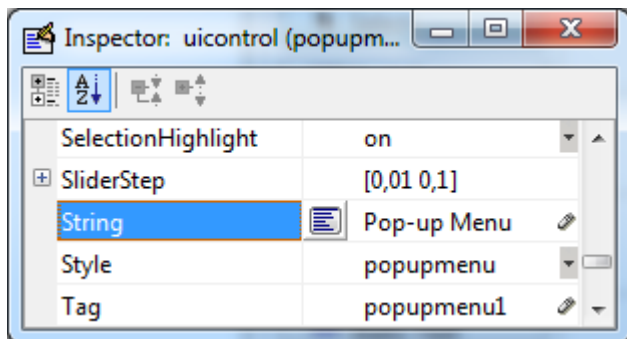


- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Feliratozzuk a komponenseket:

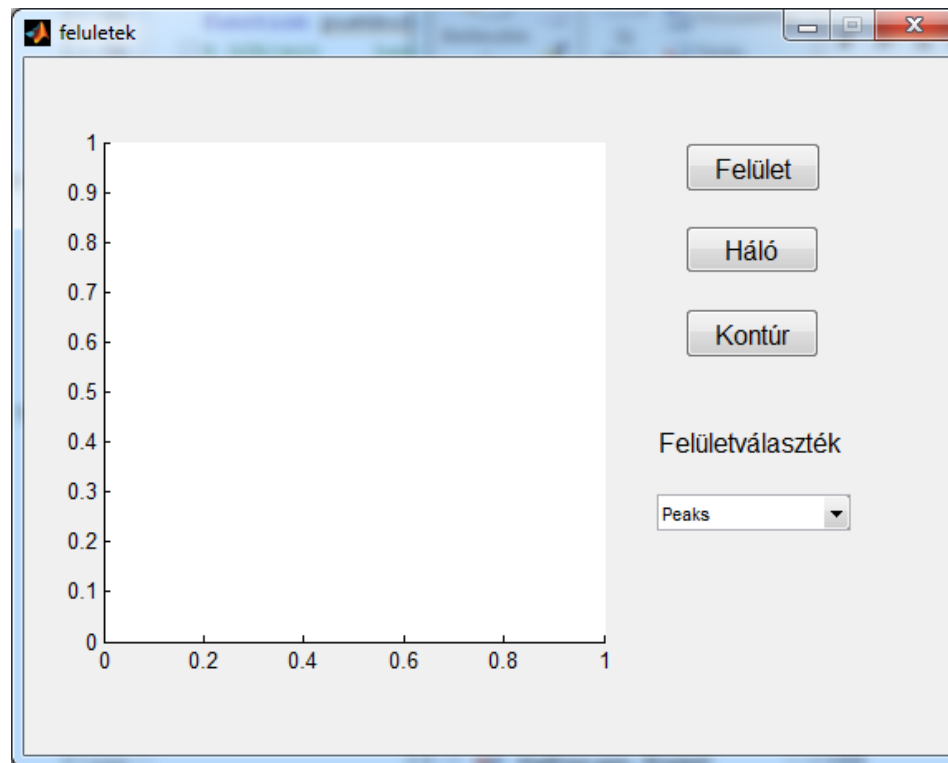


- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

- Adjuk meg a legördülő lista elemeit: **String** mellett nyomjunk a lista ikonra! A megjelenő listaeditorban adjuk meg a listaelemeket: **Peaks, Membrane, Sinc**



- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Miután elkészültünk az elrendezéssel, futtassuk az elmentéshez az alkalmazásunkat! Adjunk neki **feluletek.m** fájlnevet!
- A futó, de még érzéketlen alkalmazás:



- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Az automatikusan generált kód: 117 sor, nagyrészt komment.

```

1 function varargout = feluletek(varargin)
2 % FELULETEK MATLAB code for feluletek.fig
3 % FELULETEK, by itself, creates a new FELULETEK or raises the existing
4 % singleton*.
5 %
6 % H = FELULETEK returns the handle to a new FELULETEK or the handle to
7 % the existing singleton*.
8 %
9 % FELULETEK('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in FELULETEK.M with the given input arguments.
11 %
12 % FELULETEK('Property','Value',...) creates a new FELULETEK or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before feluletek_OpeningFcn gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to feluletek_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help feluletek
24
25 % Last Modified by GUIDE v2.5 15-Apr-2015 07:51:35
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name',       mfilename, ...
30                  'gui_Singleton',   gui_Singleton, ...
31                  'gui_OpeningFcn', @feluletek_OpeningFcn, ...
32                  'gui_OutputFcn',  @feluletek_OutputFcn, ...
33                  'gui_LayoutFcn',  [] , ...
34                  'gui_Callback',    []);
35
36 if nargin && ischar(varargin{1})
37     gui_State.gui_Callback = str2func(varargin{1});
38 end
39
40 if narginout
41     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
42 else
43     gui_mainfcn(gui_State, varargin{:});
44 end
45 % End initialization code - DO NOT EDIT
46
47 % --- Executes just before feluletek is made visible.
48 function feluletek_OpeningFcn(hObject, eventdata, handles, varargin)
49 % This function has no output args, see OutputFcn.
50 % hObject handle to figure
51 % eventdata reserved - to be defined in a future version of MATLAB
52
53 % handles structure with handles and user data (see GUIDATA)
54 % varargin command line arguments to feluletek (see VARARGIN)
55
56 % Choose default command line output for feluletek
57 handles.output = hObject;
58
59 % Update handles structure
60 guidata(hObject, handles);
61
62 % UIWAIT makes feluletek wait for user response (see UIRESUME)
63 % uiwait(handles.figure1);
64
65 % --- Outputs from this function are returned to the command line.
66 function varargout = feluletek_OutputFcn(hObject, eventdata, handles)
67 % varargin cell array for returning output args (see VARARGIN);
68 % hObject handle to figure
69 % eventdata reserved - to be defined in a future version of MATLAB
70 % handles structure with handles and user data (see GUIDATA)
71
72 % Get default command line output from handles structure
73 varargout{1} = handles.output;
74
75 % --- Executes on button press in pushbutton1.
76 function pushbutton1_Callback(hObject, eventdata, handles)
77 % hObject handle to pushbutton1 (see GCBO)
78 % eventdata reserved - to be defined in a future version of MATLAB
79 % handles structure with handles and user data (see GUIDATA)
80
81 % --- Executes on button press in pushbutton2.
82 function pushbutton2_Callback(hObject, eventdata, handles)
83 % hObject handle to pushbutton2 (see GCBO)
84 % eventdata reserved - to be defined in a future version of MATLAB
85 % handles structure with handles and user data (see GUIDATA)
86
87 % --- Executes on button press in pushbutton3.
88 function pushbutton3_Callback(hObject, eventdata, handles)
89 % hObject handle to pushbutton3 (see GCBO)
90 % eventdata reserved - to be defined in a future version of MATLAB
91 % handles structure with handles and user data (see GUIDATA)
92
93 % --- Executes on selection change in popupmenu1.
94 function popupmenu1_Callback(hObject, eventdata, handles)
95 % hObject handle to popupmenu1 (see GCBO)
96 % eventdata reserved - to be defined in a future version of MATLAB
97 % handles structure with handles and user data (see GUIDATA)
98
99 % Hint: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents a
100 % contents(get(hObject,'Value')) returns selected item from popupmenu1
101
102 % --- Executes during object creation, after setting all properties.
103 function popupmenu1_CreateFcn(hObject, eventdata, handles)
104 % hObject handle to popupmenu1 (see GCBO)
105 % eventdata reserved - to be defined in a future version of MATLAB
106 % handles empty - handles not created until after all CreateFcns called
107
108 % Hint: popupmenu controls usually have a white background on Windows.
109 % See ISPC and COMPUTER.
110 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackg
111     set(hObject,'BackgroundColor','white');
112 end

```

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

- Komment nélkül sokkal kisebb és sokkal érthetlenebb:

```
function varargout = feluletek(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @feluletek_OpeningFcn, ...
                  'gui_OutputFcn',  @feluletek_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function feluletek_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = feluletek_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
function pushbutton2_Callback(hObject, eventdata, handles)
function pushbutton3_Callback(hObject, eventdata, handles)
function popupmenu1_Callback(hObject, eventdata, handles)
function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

- Szúrjuk be az alkalmazás megjelenítése előtt lefutó `function feluletek_OpeningFcn(hObject, eventdata, handles, varargin)` iniciáló függvénybe a felületek adatait előállító kódot:

```
% ADATOK ELKÉSZÍTÉSE A RAJZOLÁSHOZ:  
handles.peaks=peaks(35);  
handles.membrane=membrane;  
[x,y] = meshgrid(-8:.5:8);  
r = sqrt(x.^2+y.^2) + eps;  
sinc = sin(r)./r;  
handles.sinc = sinc;  
handles.current_data = handles.peaks;  
surf(handles.current_data)
```

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Szúrjuk be **surf**, **mesh**, **contour** rajzoltatását indító **Felület**, **Háló**, **Kontúr** nyomógombok eseménykezelő függvényébe a rajzoltatást az aktuális adatokkal kiváltó sort:

```
% --- A FELÜLET NYOMÓGOMB MEGNYOMÁSAKOR HAJTÓDIK VÉGRE:
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)
% A surf RAJZOLÁSA AZ AKTUÁLISAN KIVÁLASZTOTT ADATOKKAL
surf(handles.current_data);
```

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

- A mesh és contour esetén is:

```
% --- Executes on button press in pushbutton2.  
function pushbutton2_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton2 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
% A mesh RAJZOLÁSA AZ AKTUÁLISAN KIVÁLASZTOTT ADATOKKAL  
mesh(handles.current_data);  
  
% --- Executes on button press in pushbutton3.  
function pushbutton3_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton3 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
% A contour RAJZOLÁSA AZ AKTUÁLISAN KIVÁLASZTOTT ADATOKKAL  
contour(handles.current_data);
```

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
- Szúrjuk be a legördülő menü kiválasztásának megváltozása esetén végrehajtandó sorokat:

```
% --- 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)  
% BESZÚRT SOROK A VÁLASZTÁS LEKEZELÉSÉRE:  
val = get(hObject, 'Value'); % LISTAELEM INDEX LEKÉRDEZÉSE  
str = get(hObject, 'String'); % A string-beli VEKTOR LEKÉRDEZÉSE  
switch str{val};  
case 'Peaks' % FELHASZNÁLÓ VÁLASZTÁSA: peaks  
    handles.current_data = handles.peaks;  
case 'Membrane' % FELHASZNÁLÓ VÁLASZTÁSA: membrane  
    handles.current_data = handles.membrane;  
case 'Sinc' % FELHASZNÁLÓ VÁLASZTÁSA: sinc  
    handles.current_data = handles.sinc;  
end  
guidata(hObject, handles)|
```

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**
 - Végül rajzoltassuk a felületeket a beállított adatokkal:

```
% --- A figure1 KREÁLÁSA ALATT HÍVÓDIK MEG, MIUTÁN MINDEN PARAMÉTERT BEÁLLÍTOTTUNK  
function figure1_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to figure1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

- A futtatás előtt ne felejtsük el lementeni a kibővített alkalmazást!

- **Egy összetettebb példa grafikus felhasználói felület (GUI) létrehozására**

- **Végre kipróbálhatjuk:**

