Obrázky v MATLAB-e GUI

Zobrazovanie obrázkov

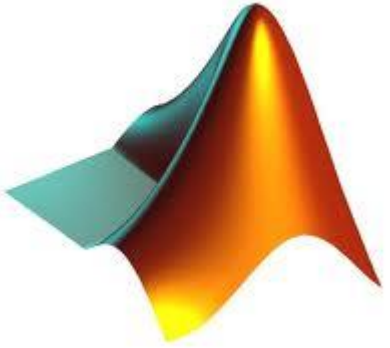
`image(M);`

- priamo v matlabe
- farby zobrazovaného obrazu vôbec nemusia zodpovedať reálnym farbám

`imshow(M);`

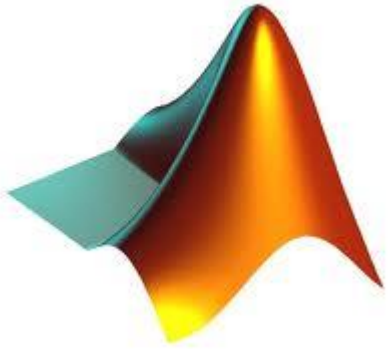
- IPT
- predpokladá, že zobrazované hodnoty sú intenzity pixlov

- `figure;`



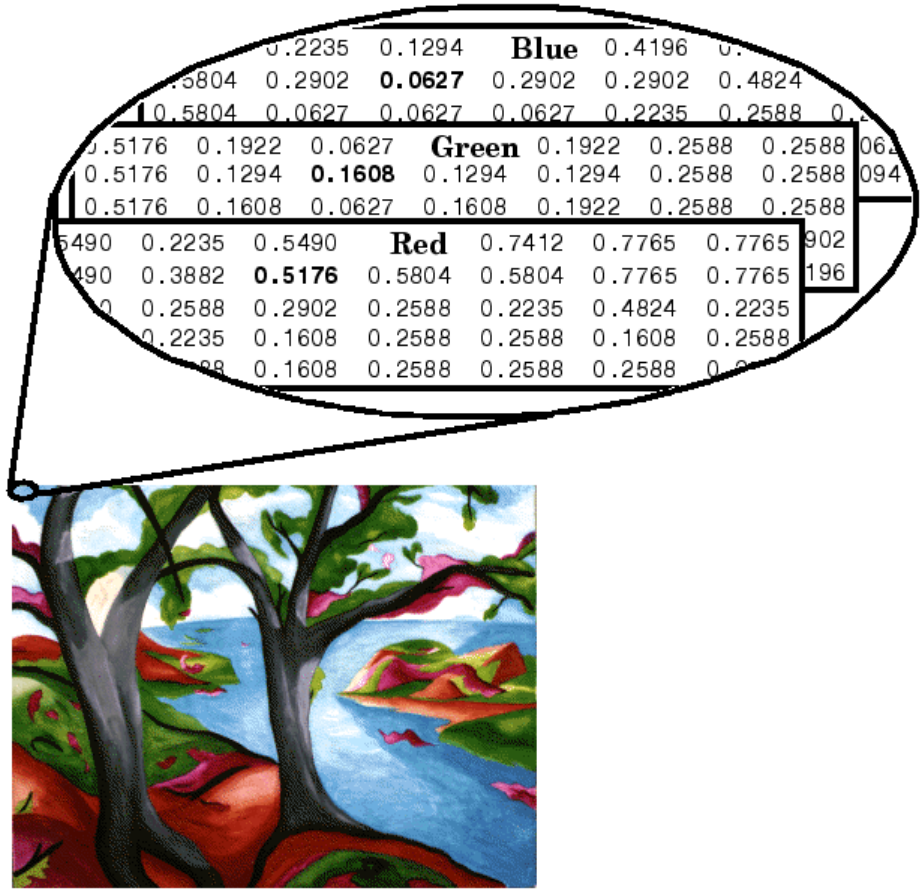
Zobrazovanie obrázkov - rozdiel

```
img = imread('cameraman.tif');

figure;

image(img);

set(gcf,'colormap',gray);

figure;

subplot(1,2,1);

image(img); %axis off; axis image;

subplot(1,2,2);

imshow(img);
```



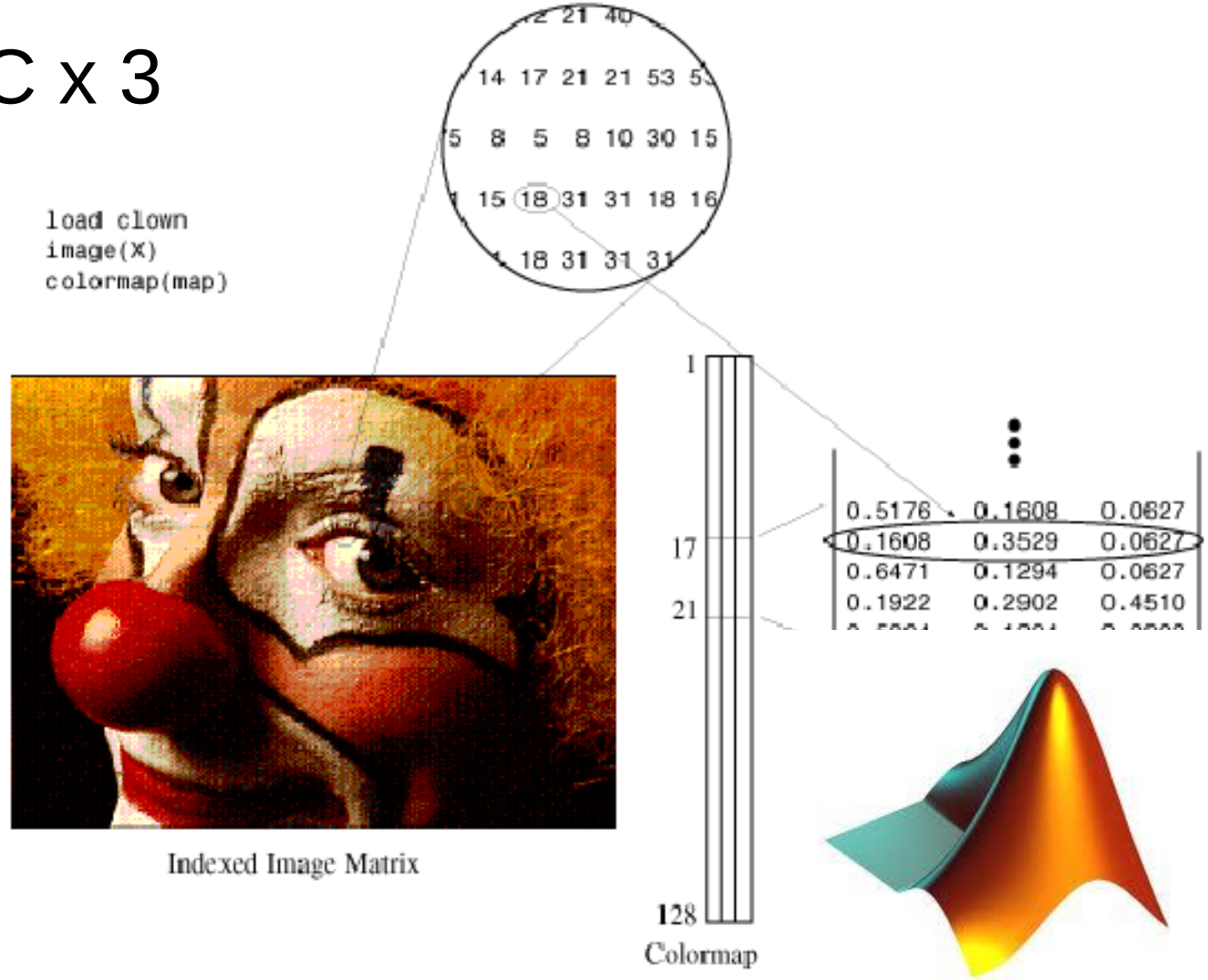
True Color vs. Indexed Images

- True color:
  - obrázok veľkosti MxN je uchovaný v 3-rozmernom poli
  - M x N x 3 (RGB hodnoty)



Indexed image

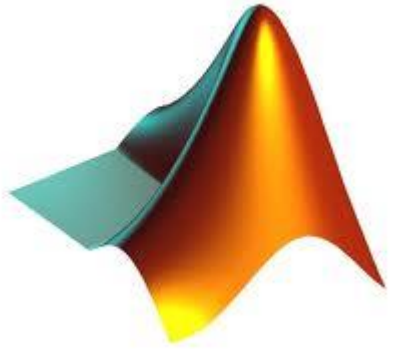
- obrázok M x N je uchovaný v matici M x N
- Farby v matici C x 3



Colormap

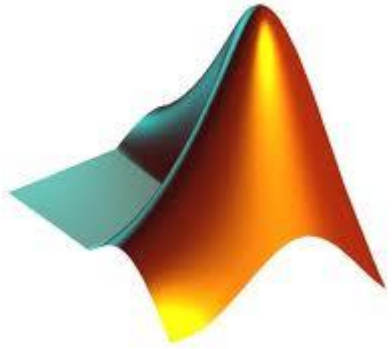
- hodnoty v intervale [0,1]
- colormap(map);
- colormap(hsv(128));


```
load clown
imshow(X,colormap(map));
imshow(X,colormap(jet));
imshow(X,colormap(spring));
```

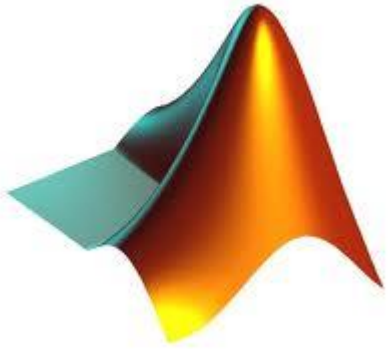


Konverzie

```
im = imread('nazov.jpg');
image(im);
% nastav C<=65536
C=16;
[X,map] = rgb2ind(im, C);
 load clown
 imshow(X,colormap(map));
  RGB = ind2rgb(X,map);
```



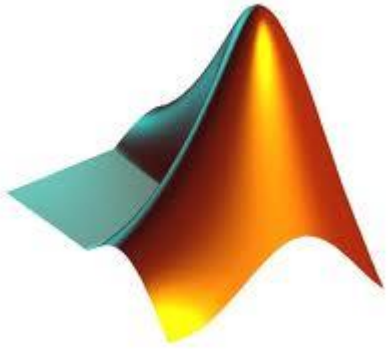
- MATLAB podporuje formáty
  - BMP, JPG, PNG, TIFF, GIF
  - JPEG 2000 formáty: JP2, JPX...
  - Iné: PNM, PCX, ICO, PBM, HDF...



M-files

- MATLAB skript
- postupnosť príkazov

- MATLAB funkcia
- meno súboru = názov funkcie
- `prvy.m`

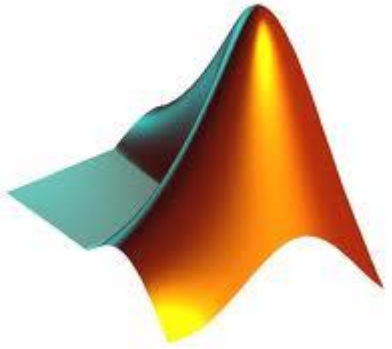


M-files

```
function x = prvy (v)
x = v(1);
```

- Volanie:

```
y = [2 4 6 7 5 9];
x = prvy(y);
```

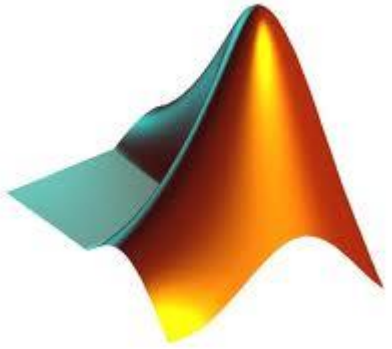


M-files

- `function [x,y,z] = prvy(v)`
- `prvy(v);`
- uloží do ans len x

- `function [] = prvy(v)`

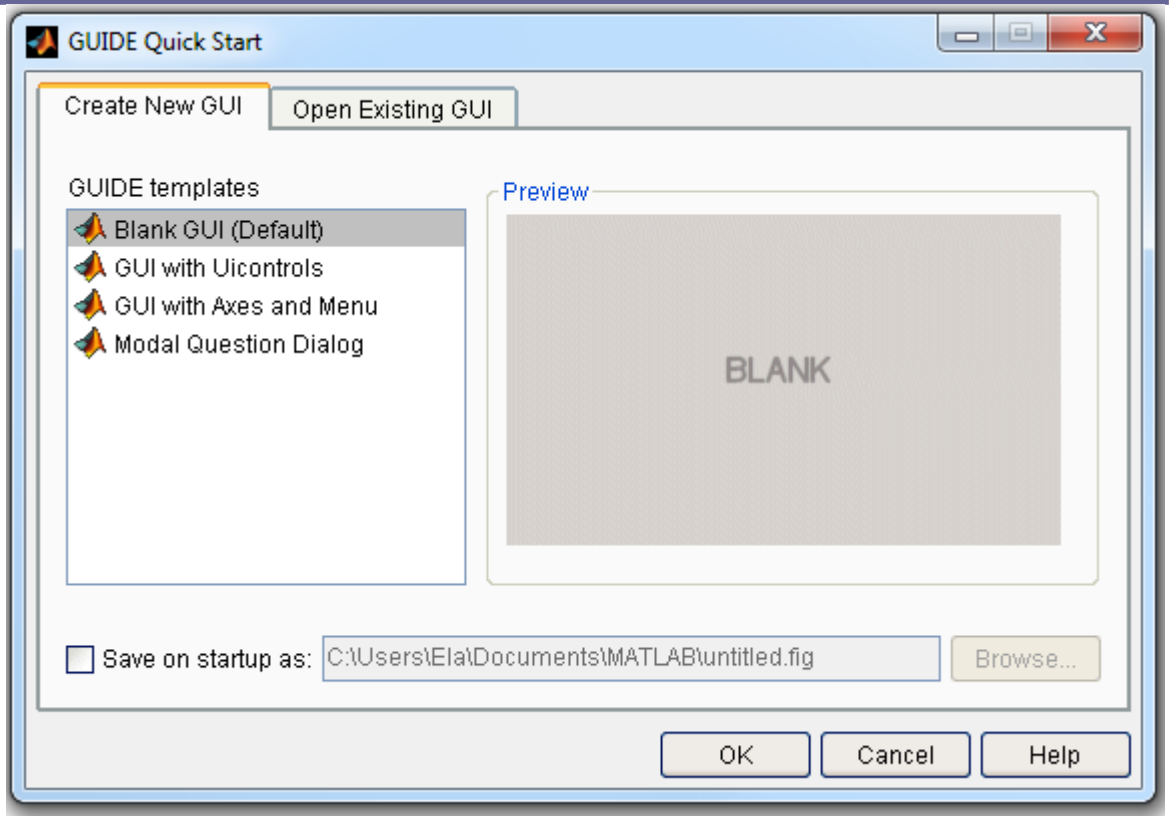
- `% komentare`
- `%% spúšťateľná časť kódu`



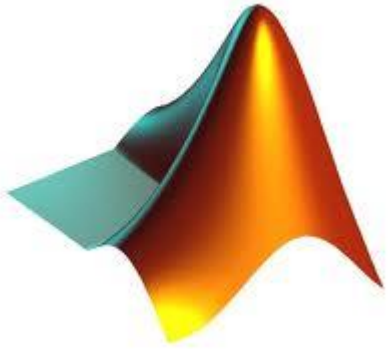
GUI

```
>> guide
```

- Blank GUI

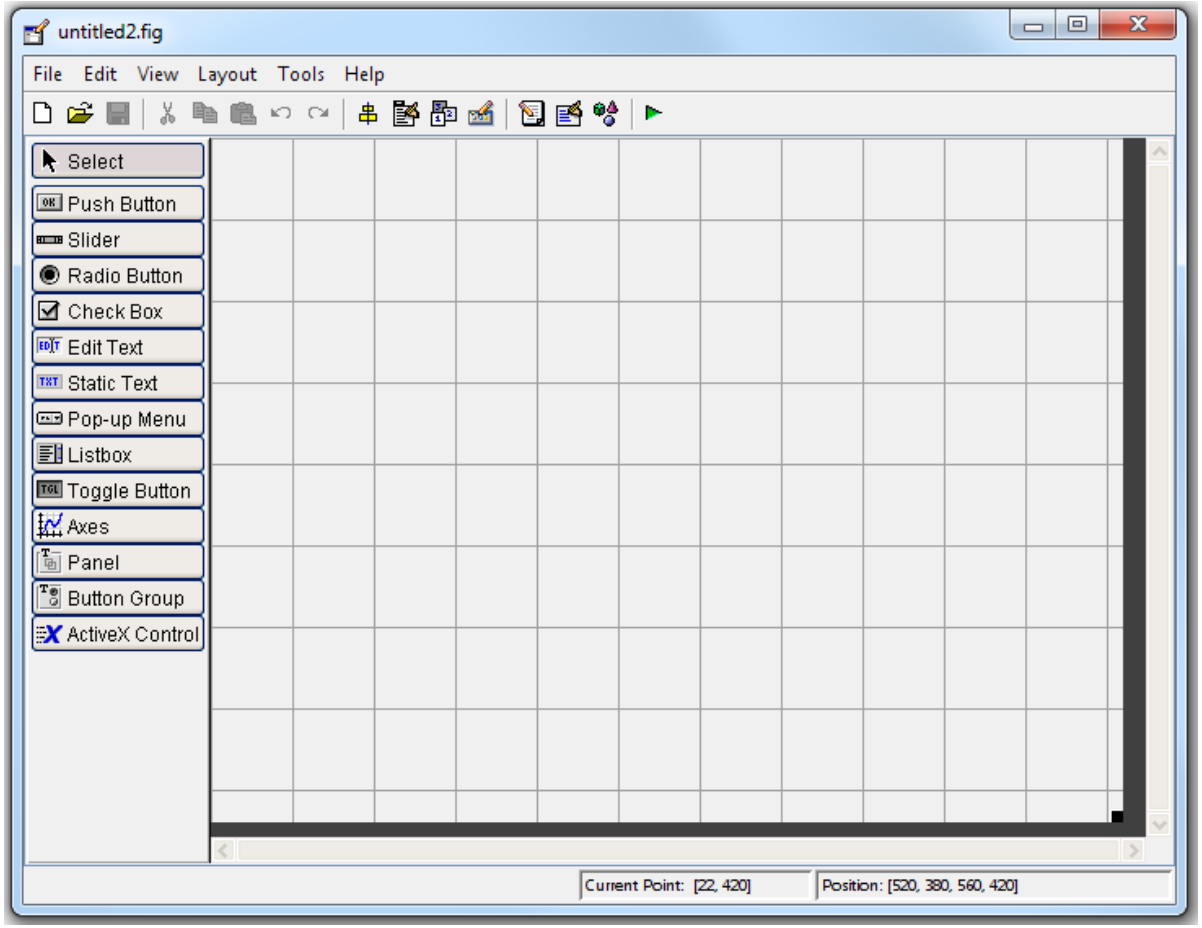


- Vytvorí dva súbory:
  - `meno.fig`
  - `meno.m`
  - Nemeniť meno už vytvoreného GUI



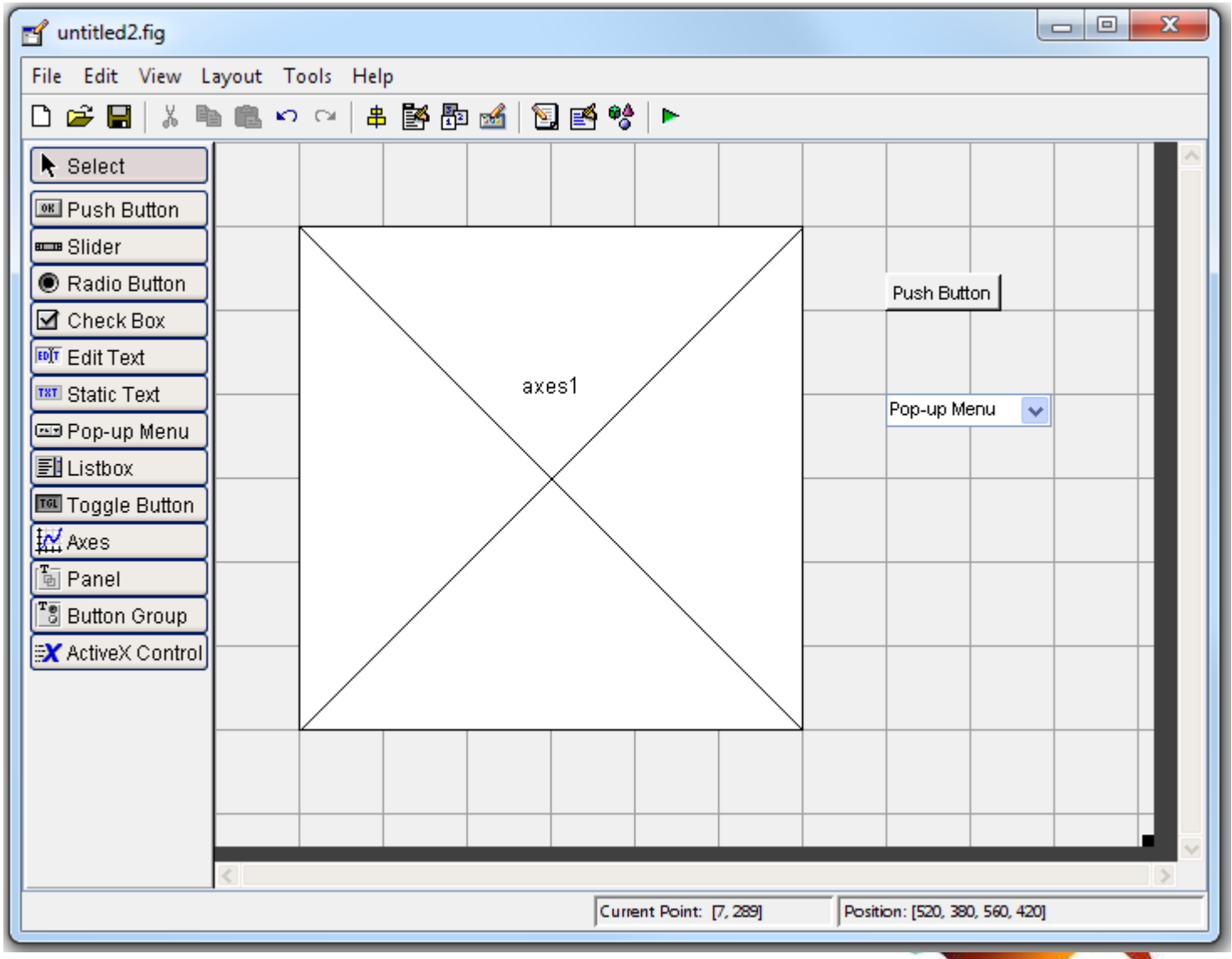
GUI

- GUI objekty:
  - Button, radio button, check box, slider
  - Edit text, Static text
  - Axes
  - Pop-up menu
  - List box
  - Panel
  - Button group...



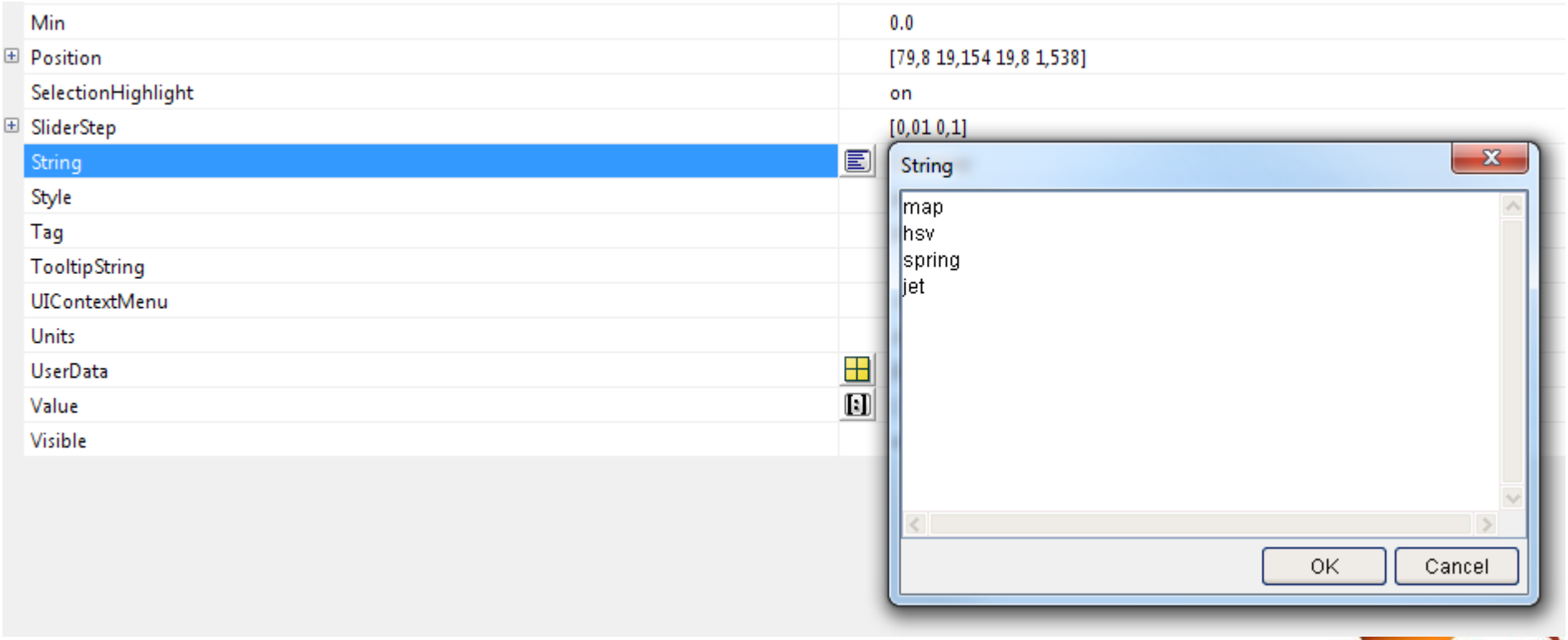
GUI

- Načítať
- Vykresliť
- Zmeniť farby



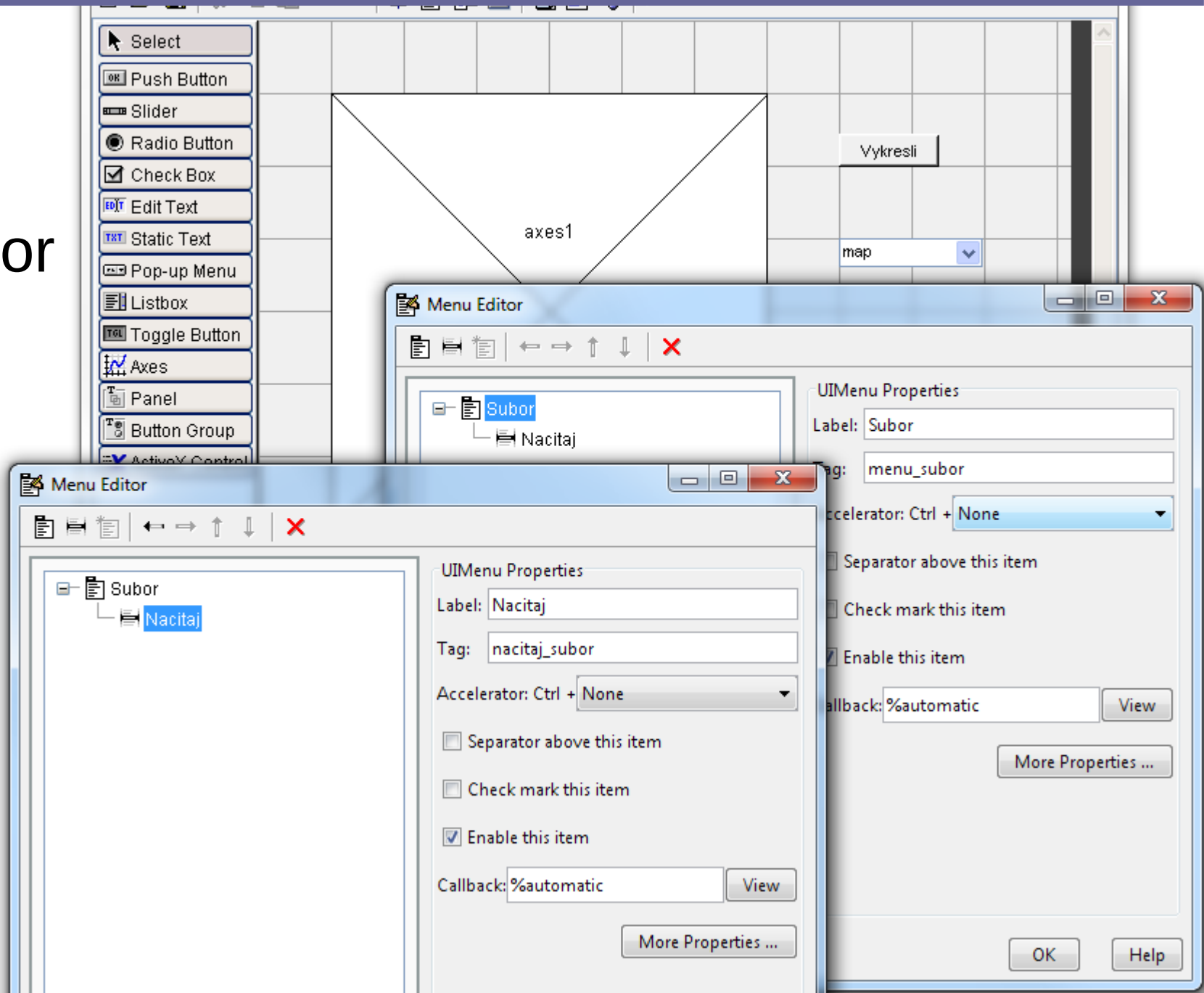
GUI

- Property Inspector
  - Color, text, name, position, opacity



GUI

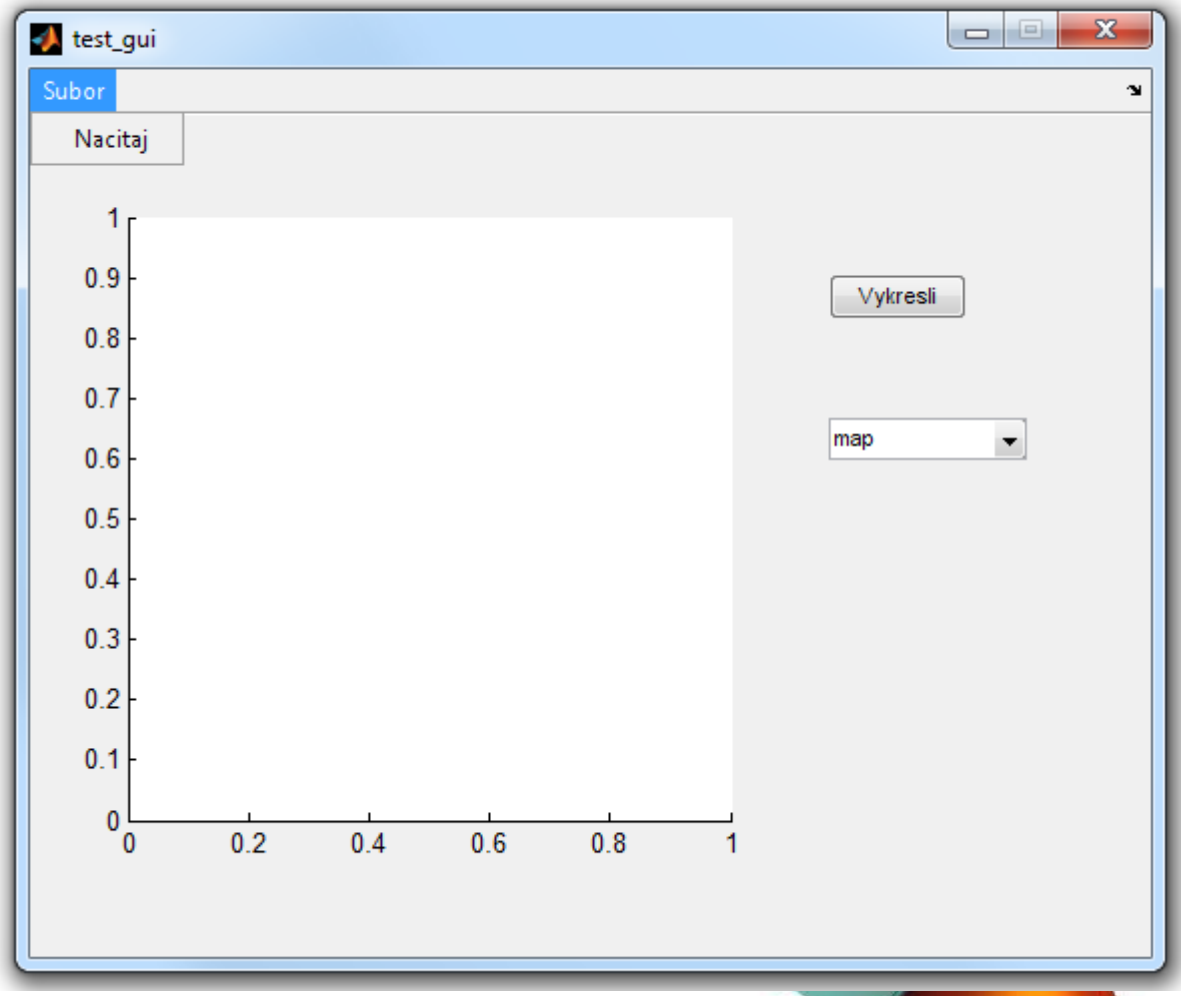
- Tools
- Menu Editor



GUI

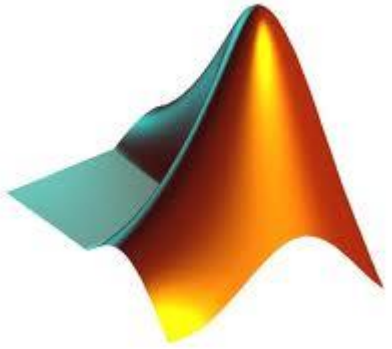
- Save as
- Spusti
- Nic sa nedeje!
- Treba dopísať kód

```
>> test_gui
>>
```



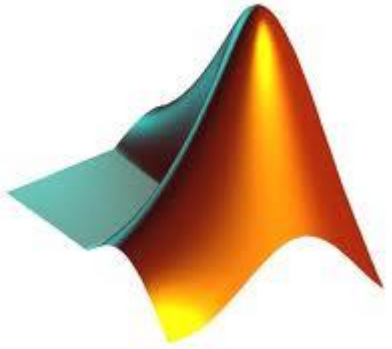
GUI

- Callbacks = funkcie, ktoré sa vykonajú po aktivácii objektu
- ak chceme využívať v jednom callbacku premennú ktorú sme vytvorili v inom, použijeme funkcie get a set



GUI - Callbacks

- Callback
- ButtonDownFcn
- KeyPressFcn
- CreateFcn
- ...



GUI

- Handles = štruktúra uchovávajúca data

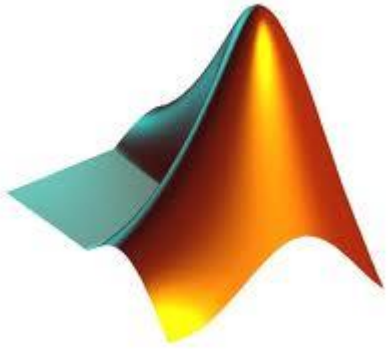
```
set(handles.text2,'Visible','on');
g = get(handles.radiobutton1, 'Value');
```

set(object, 'property',value)

get(object, 'property')

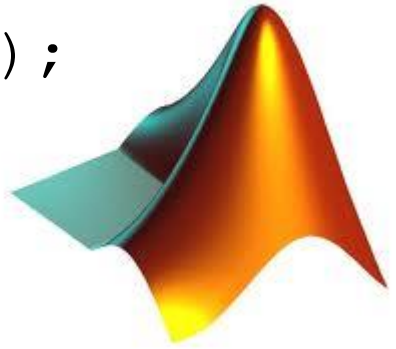
- Globálne data:

```
handles.moje_data = hodnota;
guidata(hObject,handles)
```



GUI

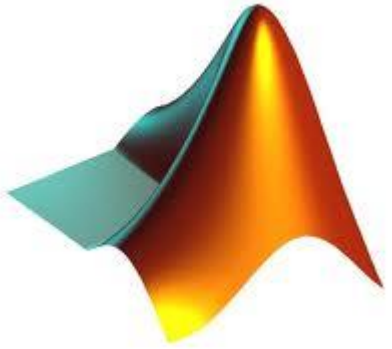
- Objekty majú okrem štandardných parametrov tzv. `'UserData,`
  - môžme vložiť ľubovoľné dáta (obrázok, číslo)
- V jednom callbacku načítame
  - `RGB = imread('1.jpg');`
  - `set(handles.pushbutton1,'UserData',RGB);`
- V druhom zavoláme
  - `I=get(handles.pushbutton1,'UserData');`
  - `imshow(I);`



GUI

- Načítanie obrázka

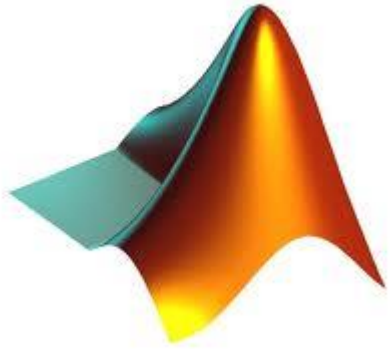
```
[FileName,PathName]=uigetfile('*.jpg',
'Vyber .jpg');

I = imread(fullfile(PathName,
FileName));

figure; imshow(I);
```



GUI

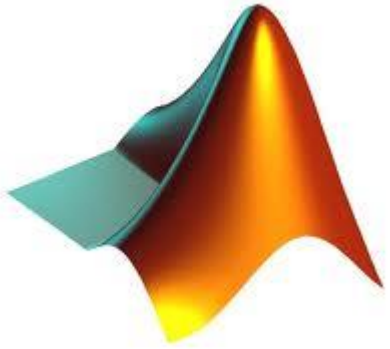
```
function test_gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command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

```
% Create color maps
handles.map=colormap(jet);
handles.hsv=colormap(hsv(128));
handles.spring=colormap(spring);
handles.jet=colormap(jet);
% Set the current map value
handles.current_map = handles.map;
```



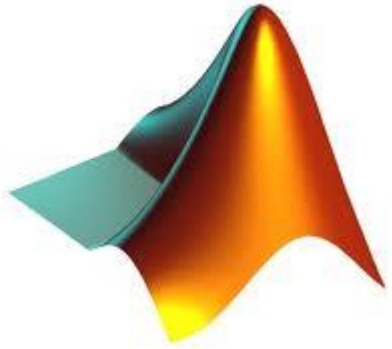
GUI

```
function nacitaj_subor_Callback(hObject, eventdata, handles)
% hObject    handle to nacitaj_subor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 [i_file,i_PathName] = uigetfile({'*.jpg', 'JPEG imagefile
(*.jpg)'; '*.*', 'All Files (*.*)'}, 'Select the JPEG
Image',[cd '\']);
if ~isequal(i_file, 0)
    % Reading the Image file
    i_file = fullfile(i_PathName,i_file);
    i_RGB = double(imread(i_file))/255;
    [idx_im,handles.map] = rgb2ind(i_RGB, 256);
    handles.index_image=idx_im;
    handles.current_map = handles.map;
end
```



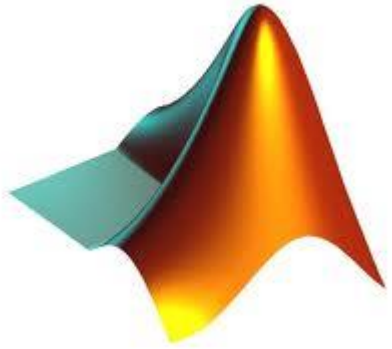
GUI

```
% Reset PopUp menu to 1st color map
set(handles.popupmenu1,'Value',1)

% Save the handles structure.
guidata(hObject,handles)
```



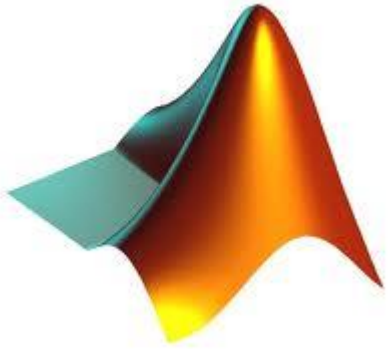
GUI

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

% Determine the selected color map
str = get(hObject,'String');
val = get(hObject,'Value');
```



GUI

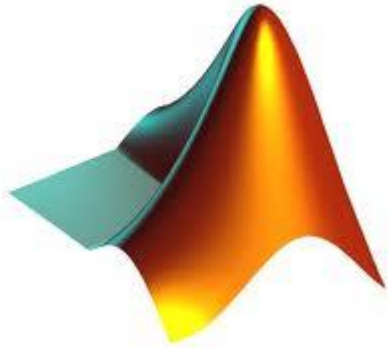
```
% Set current data to the selected data set.
switch str{val};
case 'map'
   handles.current_map = handles.map;
case 'hsv'
   handles.current_map = handles.hsv;
case 'spring'
   handles.current_map = handles.spring;
case 'jet'
   handles.current_map = handles.jet;
end
colormap(handles.current_map)
% Save the handles structure.
guidata(hObject,handles)
```



GUI

```
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)
```

```
imshow(handles.index_image)
colormap(handles.current_map)
```



Tutoriál na doma

- http://www.mathworks.com/matlabcentral/fileexchange/27773-matlab-video-tutorial-in-czech-lesson-12--creating-gui