

# Grafická uživatelská rozhraní v MATLABu

(Pomocný text verze 0.2 část 4 — 15.3.2004 — Michal Hajžman)

---

Tento materiál obsahuje velice stručné a subjektivně vybrané informace potřebné pro vytváření grafických uživatelských rozhraní v MATLABu. Měl by sloužit jako přehled při cvičeních a také jako pomůcka při samostatném studiu :-). Pro další studium doporučuji elektronickou dokumentaci *Creating Graphical User Interfaces*.

V přehledu jsou ve většině případů uváděny příkazy bez parametrů, přestože vstupní či výstupní parametry existují. Jestliže není uveden dostatečně názorný příklad pro použití příkazu, lze pomocí příkazu `help <jmeno_prikazu>` získat okamžitou podrobnou nápovědu.

## 1 Obecně

Následující informace jsou opravdu jen velmi stručným úvodem do problematiky vytváření GUI v MATLABu.

GUI se vytváří v grafickém okně, skládá se z různých ovládacích prvků (tlačítka, seznamy, editace, ...), které budeme nazývat objekty. Objekty budeme nazývat také všechny ostatní grafické objekty (osy, čáry, plochy, ...). Každý objekt má své jedinečné číslo (*handle*) a své vlastnosti. Pomocí funkce `get` lze vlastnosti objektů zjišťovat a pomocí funkce `set` je lze nastavovat na možné hodnoty. Tyto dvě funkce jsou pro nás velmi důležité.

Při vytváření GUI můžeme postupovat dvěma způsoby.

- (a) Použijeme vývojové prostředí GUIDE. Nemusíme umět tolik programovat, použití je „wokení a myšoidní“.
- (b) Celé naše GUI naprogramujeme uvnitř jedné funkce.

Přímo v MATLABu lze získat informace pomocí příkazů nápovědy `help graphics` a `help uitools`.

## 2 CALLBACKy

Objekty mají vlastnost označenou `Callback`, která obsahuje akci, jenž se provádí po výběru objektu (například zmáčknutí tlačítka). Akce může být jednoduchý příkaz MATLABu, ale zpravidla je akce zapsána ve funkci a vlastnost `Callback` je nastavena na řetězec určující jméno funkce s případnými parametry.

## 3 GUIDE

Funkce `guide` spouští vývojové prostředí pro vytváření GUI. Zde můžeme vybrat a rozmístit požadované ovládací a grafické objekty. Nastavovat a měnit jejich vlastnosti. GUIDE vytváří soubor s příponou `fig`, kde jsou zapsány informace o grafickém okně, a soubor s příponou `m`, který lze editovat a programovat tak „callbacky“ jednotlivých funkcí.

## 4 Funkce pro vytváření objektů

<code>menu</code>	Vytváří samostatné menu.
<code>uimenu</code>	Vytváří menu v horní liště okna.
<code>uicontextmenu</code>	Vytváří kontextové menu.
<code>uicontrol</code>	Asi nejdůležitější vytvářecí funkce, pomocí níž lze vytvářet všechny možné ovládací prvky.

## 5 Ruční programování

Jestliže chceme kompletní GUI naprogramovat sami, je dobré dodržovat určité zásady při vytváření funkce s vlastním GUI. Vhodné je použití struktury `switch` pro rozdělení požadovaných akcí. Následuje jednoduchá ukázka.

## 6 Příklad

```
% PRIKLAD Ukazkový program pro vytváření GUI.
% ~~~~~
%
% Volání:      priklad(akce)
% ~~~~~
%
%             IN: akce ... retezec urcujici akci :-)
%
% Mozné akce:  init ... inicializace
% ~~~~~       osy .... zapne/vypne osy
%             cmap ... prepne schema
%
% Poznámky:    Funkce zobrazí osy a vykreslí kouli.
% ~~~~~
%             Tlacička dovolí prepínat vzhled koule.

function priklad(akce)

switch akce

% *****
% * akce init *
% *****
case 'init'

% Vytvoření okna
% ~~~~~
figure('NumberTitle', 'off', ...
      'Name', 'Ukazkové GUI', ...
      'Units', 'normalized', ...
      'Position', [ 0.2 0.2 0.6 0.6 ] );

% Vytvoření os a vykreslení koule
% ~~~~~
axes('Units', 'normalized', ...
     'Position', [ 0.1 0.1 0.6 0.8 ] );
[X,Y,Z] = sphere(30);
h_koule = surf(X,Y,Z);
axis equal

% Tlacičko osy
% ~~~~~
h_tl_osy = uicontrol('Style', 'pushbutton', ...
                    'String', 'Osy ZAP/VYP', ...
                    'Callback', 'priklad(''osy'')', ...
                    'Units', 'normalized', ...
```

```

        'Position', [ 0.8 0.7 0.15 0.05 ] );

% PopUpMenu colormap + popis
% ~~~~~~
uicontrol('Style', 'text', ...
    'BackgroundColor', [0.8 0.8 0.8], ...
    'String', 'Barevne schema:', ...
    'Units', 'normalized', ...
    'Position', [ 0.8 0.48 0.15 0.05 ] );

h_lb_cmap = uicontrol('Style', 'popupmenu', ...
    'String', 'hsv|gray|copper|autumn|cool|hot', ...
    'Callback', 'priklad(''cmap'')', ...
    'Units', 'normalized', ...
    'Position', [ 0.8 0.4 0.15 0.05 ] );

% *****
% * akce osy *
% *****
case 'osy'

    % Vypne nebo zapne osy
    % ~~~~~~
    stav = get(gca,'Visible');
    if strcmp(stav,'on')
        axis off
    else
        axis on
    end

% *****
% * akce cmap *
% *****
case 'cmap'

    % Prepne barevne schema
    % ~~~~~~
    schema = get(gcbo,'Value');
    switch schema
        case 1, colormap(hsv)
        case 2, colormap(gray)
        case 3, colormap(copper)
        case 4, colormap(autumn)
        case 5, colormap(cool)
        case 6, colormap(hot)
    end

end

end

```

## 7 Závěrem

V příkladu nebyla využita práce s čísly objektů (handly), protože to k řešení nebylo potřeba. V praktických GUI tomu je většinou právě naopak, handly je nutno využívat velice často. Pro práci s handly existují vhodné funkce. Nejjednodušší je využití funkce `findobj` společně s vhodně nastavenými „tagy“ objektů. Vše je popsáno v nápovědě či v elektronické dokumentaci, kterou opět vřele doporučuji.