

MATLAB®

Grafické uživatelské rozhraní



Tato elektronická publikace byla zakoupena u: Karel Zaplatílek,
slunce007@seznam.cz

Jméno a příjmení kupujícího: **Karel Zaplatílek**
E-mail: **slunce007@seznam.cz**

Upozorňuji, že elektronická kniha je dílem chráněným podle autorského zákona a je určena jen pro osobní potřebu kupujícího. Kniha jako celek ani žádná její část nesmí být volně šířena na internetu, ani jinak dále zveřejňována.

OBSAH

Předmluva	4
1 Úvod	5
2 Tvorba GUI ručním způsobem	6
2.1 Grafické objekty <i>uicontrol</i> , tvorba tlačítka	6
2.2 Hierarchie grafických objektů v MATLAB. Grafické objekty <i>root</i> , <i>figure</i> a <i>axes</i>	15
2.3 Tvorba přepínače <i>toggle</i>	23
2.4 Tvorby zatržitek <i>radiobutton</i> a <i>checkbox</i>	24
2.5 Tvorba grafických objektů <i>edit</i> a <i>text</i>	26
2.6 Tvorba grafického objektu <i>slider</i>	27
2.7 Tvorba grafického objektu <i>frame</i>	28
2.8 Tvorba grafického objektu <i>listbox</i>	29
2.9 Tvorba grafického objektu <i>popupmenu</i>	31
2.10 Tvorba grafických objektů <i>uimenu</i> , <i>uipanel</i> a <i>uitable</i>	31
2.11 Datová schránka <i>UserData</i>	34
2.12 Další užitečné tipy.....	36
2.13 Technika <i>Switched Board Programming</i>	37
2.14 Praktický příklad aplikace s GUI.....	40
3 Tvorba GUI aplikací <i>App Designer</i>.....	46
3.1 Spuštění prostředí a grafický návrh aplikace	46
3.2 Programování událostí a dalších funkcionalit	50
3.3 Výpis klíčových částí zdrojového kódu	58
4 Závěr	60
Odkazy.....	61

Předmluva

Vážený čtenáři, tato kniha je učebnicí základů tvorby grafického uživatelského rozhraní v programovém prostředí MATLAB[®], známé pod zkratkou GUI (*Graphical User Interface*). Je to vrstva, kterou uživatel ovládá svůj počítačový algoritmus. Obsahuje řadu grafických objektů jako jsou tlačítka, přepínače, různá zatržítka, pole pro vložení textu nebo jeho prostý výpis, tahové potenciometry, různá menu, tabulky, grafy a další objekty.

Tvořit GUI v prostředí MATLAB lze dvěma principiálními postupy; textovým, kdy ručně píšete zdrojové texty v rámci funkce anebo s využitím generátoru grafických objektů, kde pracujete především myší.

Jako učitel na technické fakultě univerzity a také v rámci řešení vědeckých projektů vytvářím GUI se svými studenty a spolupracovníky již řadu let. Na základě zkušeností pracuji převážně ručním textovým způsobem, neboť umožňuje mít práci plně pod kontrolou a produkuje uživatelsky optimální kód.

Od roku 2016 obsahuje MATLAB moderní aplikaci *App Designer*, která většinu práce zastane za vás. Její kód je však tvořen automatizovaně, což je na jednu stranu rychlé, avšak někdy za cenu nižší čitelnosti kódu a jeho optimálnosti. Každý si však může vybrat způsob, který mu bude vyhovovat.

V knize se podíváme na oba způsoby tvorby GUI, více prostoru je však věnováno ručnímu způsobu práce, který osobně preferuji. Pro co největší efektivitu vám doporučuji dobře zvládnout úplné základy práce v MATLAB, čemuž se věnuje má kniha pro začínající uživatele [1], viz odkazy na literaturu v závěru knihy. Doporučuji vám mít obě knihy.

Přeji si, aby se vám tato kniha stala praktickou příručkou tvorby GUI pro vaše aplikace. Poznáte, jak naplňující je tvořit nové věci a objevovat nové postupy a možnosti. Tvořit GUI v MATLAB je snadné a efektivní.

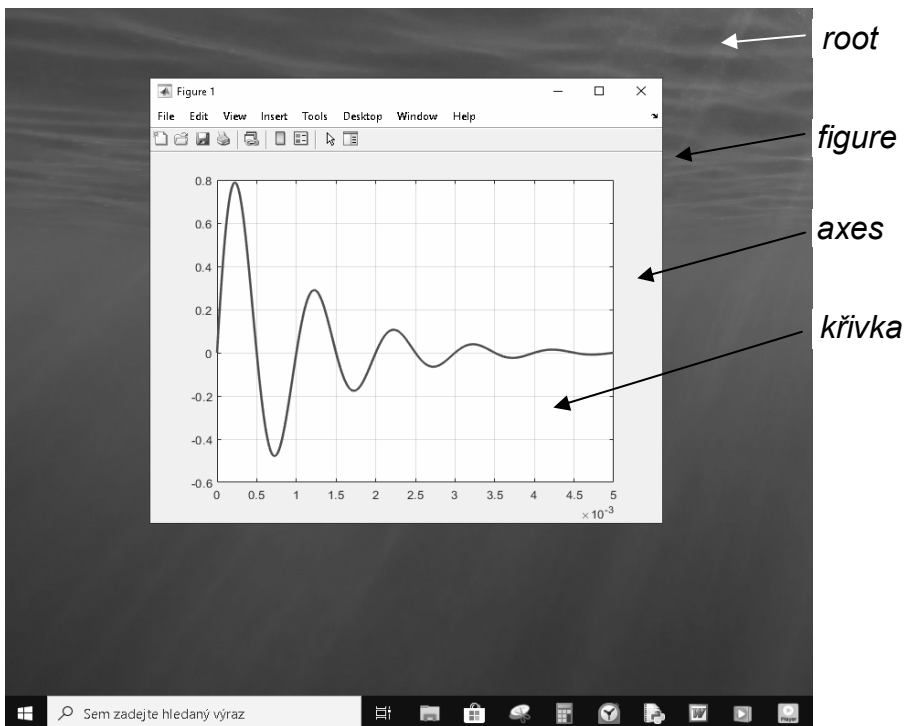
Knihu jsem napsal po mnoha letech zkušeností s tvorbou GUI jak v akademickém prostředí univerzit, tak v rámci spolupráce s výzkumnými ústavy a dalšími subjekty. Moje nabyté praktické zkušenosti jsou vtěleny v této knize. Všem, kteří k tomu přispěli, upřímně děkuji.

Brno, září 2020

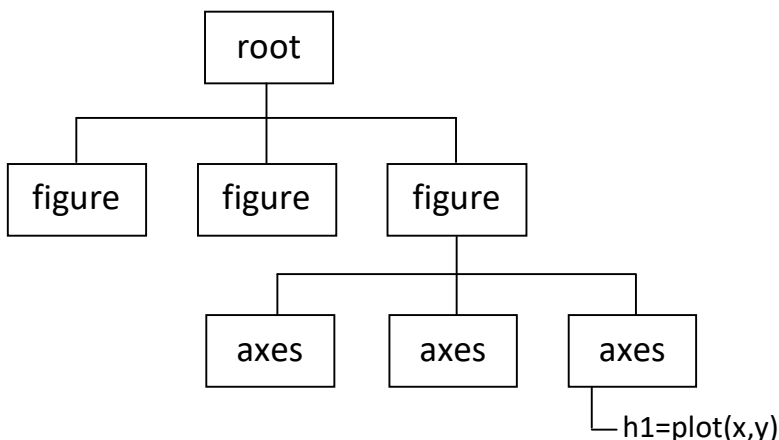
Autor

2.2 Hierarchie grafických objektů v MATLAB. Grafické objekty *root*, *figure* a *axes*

V prostředí MATLAB mají všechny grafické objekty (včetně křivek) určitou danou hierarchii, viz *obr.2.12* a *obr.2.13*.



Obr.2.12. Grafické objekty „root“, „figure“, „axes“ a „křivka“.



Obr.2.13. Hierarchie hlavních grafických objektů v MATLAB.

Nejvýše stojí grafický objekt *root*, který si představte jako obrazovku vašeho monitoru. Tento objekt *root* má unikátní *handle* nastavený na 0 (nula). Pomocí *get(0)* získáme seznam všech jeho parametrů, viz *obr.2.14*. Objekt *root* nebudeme používat příliš často, hodí se snad jen k získání informací o rozlišení monitoru či barevné hloubce, viz *obr.2.15*.

```
Command Window
>> get(0)
    CallbackObject: [0x0 GraphicsPlaceholder]
      Children: [0x0 GraphicsPlaceholder]
    CurrentFigure: [0x0 GraphicsPlaceholder]
FixedWidthFontName: 'Courier New'
  HandleVisibility: 'on'
  MonitorPositions: [1 1 1920 1080]
        Parent: [0x0 GraphicsPlaceholder]
  PointerLocation: [649 458]
        ScreenDepth: 32
ScreenPixelsPerInch: 96
        ScreenSize: [1 1 1920 1080]
  ShowHiddenHandles: off
            Tag: ''
            Type: 'root'
            Units: 'pixels'
        UserData: []
```

Obr.2.14. Seznam parametrů grafického objektu „root“.

```
Command Window
>> roz=get(0,'screensize') % dotaz na rozlišení monitoru

roz =

         1         1        1920        1080

>> bh=get(0,'screendepth') % barevná hloubka

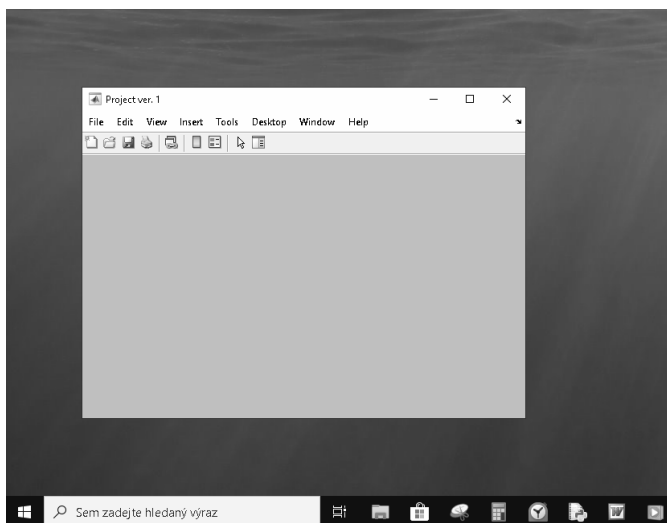
bh =

    32
```

Obr.2.15. Dotaz na rozlišení monitoru v rámci grafického objektu „root“.

Jestliže vykreslíte nějakou křivku např. příkazem *plot*, systém automaticky vytvoří nové okno, do něhož křivku nakreslí, *obr.2.12*. Neuvedete-li jinak, rozměry, barvy a další parametry budou implicitní.

```
Fcn_figure.m x Fcn_2figure.m x +
1 % hierarchie grafických objektu
2 %-----
3 %--- tvorba figure ---
4 F=figure('units','norm','position',[0.1 0.25 0.4 0.38],...
5         'numbertitle','off','color',[0.75 0.75 0.75],...
6         'name','Project ver. 1','tag','figure1');
```



Obr.2.16. Tvorba grafického objektu „figure“.

Jak vidíte na *obr.2.11*, grafický objekt *figure* můžeme tvořit záměrně a také u něj nastavovat základní parametry. Můžeme pracovat v příkazové řádce, avšak vzhledem k většímu počtu parametrů je vhodnější vtělit kód do m-souboru (scriptu nebo funkce).

Na *obr.2.16* vidíte záměrné vytvoření okna *figure*, jehož *handle* je *F*. Jsou zde nastavené jednotky na *norm* (pozice a rozměr od 0 do 1 v rámci *root* tedy vaší obrazovky počítače), dále pozice a rozměr okna (0 až 1), je vypnuto automatické generování jména okna, nastavena jeho barva, jméno a štítek *tag*.

Pozor: kód je zapsán ve scriptu, protože jak již víte, funkce nepředává do *Workspace* žádné proměnné a *handle F* by v příkazové řádce neexistoval.

Nezavírejte vytvořené okno a v příkazové řádce realizujte:

```
get(F) - seznam všech parametrů objektu figure  
b=get(F,'color')  
set(F,'color',[1 1 0])  
set(F,'menubar','none')  
set(F,'menubar','figure')  
set(F,'visible','off')  
set(F,'visible','on')  
p=get(F,'position')  
set(F,'position',[0.15 0.15 0.8 0.7])  
set(F,'units')  
delete(F) - smazání okna  
r=get(F,'parent') - vrací handle svého rodiče, tedy objektu root  
F1=get(0,'children') - vrací handle dítěte objektu root, tedy figure  
delete(get(0,'children')) - smaže vše, co je podřízeno objektu root
```

Pozor: příkazem *delete(F)* vymažete jakýkoli objekt, jehož *handle* je *F*.

Jak vidíte výše, pojmy *parent* a *children* ukazují na hierarchii grafických objektů, *obr.2.13*. Jsou platné pro všechny úrovně hierarchie. Tak např. nakreslená křivka je dítětem (*children*) kreslicího plátna *axes*, které samo je dítětem okna *figure*. Asi jako v rodině.

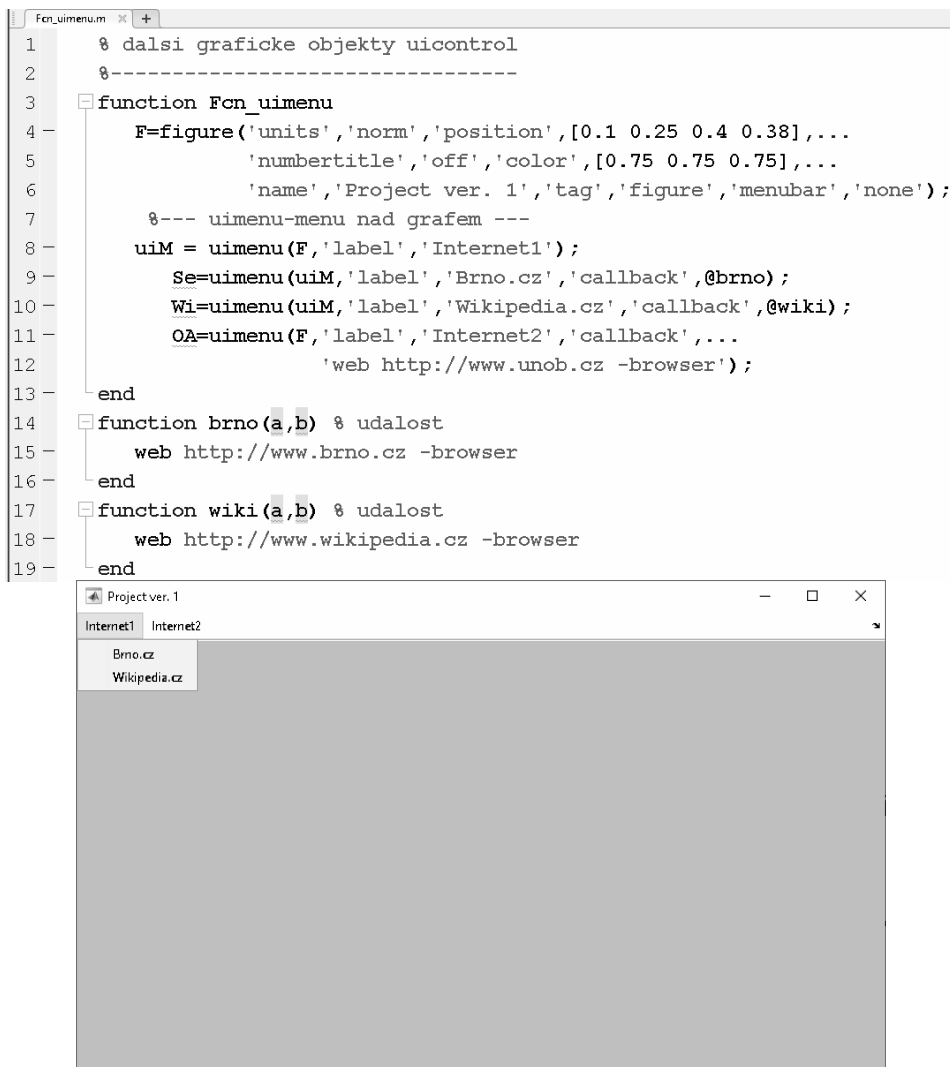
Poslední uvedený příkaz *delete(get(0,'children'))* je nejsilnějším příkazem k vymazání všech oken. Můžete jej použít např. kdyby vaše GUI zamrzlo nějakou logickou chybou. Příkaz pak zapíšete a potvrdíte v příkazové řádce.

2.10 Tvorba grafických objektů *uimenu*, *uipanel* a *uitable*

Dosud jsme pracovali s grafickými objekty typu *uicontrol*, které jsme vytvářeli příkazem *uicontrol* a rozlišovali parametrem *style*, viz jejich přehled na *obr.2.6*. Kromě nich však existuje ještě několik grafických objektů, jež se tvoří zvláštními příkazy. Podívejme se na tři z nich.

Grafický objekt *uimenu* je rozbalovací menu, které se však nevytváří uvnitř *figure*, ale na liště v horní části, viz *obr.2.27*. Mohou mít jednu nebo více úrovní; jejich událost se ošetřuje podobně vnitřními funkcemi. Je-li událost krátká, lze ji vepsat přímo do řádku vedle parametru *callback*, viz řádek č. 12.

Jsou-li parametrem *menubar* potlačena implicitní menu a tlačítka, bude zde jen naše *uimenu*.



Obr.2.27. Grafický objekt „uimenu“.

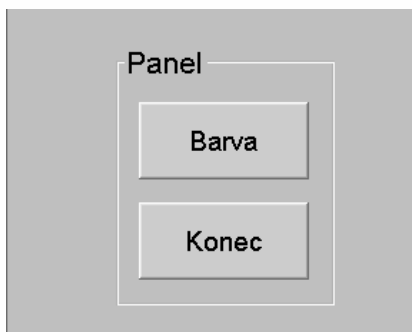
Dalším zajímavým grafickým objektem je *uipanel*. Je to v podstatě rám s možnostmi volby orámování a popisu. Jeho zvláštností však je, že může v sobě zahrnovat objekty *uicontrol*, může být jejich jakýmsi kontejnerem, viz obr.2.28. Umístíte-li *handle* objektu *uipanel* na začátek grafických objektů *uicontrol* (zde *UIP* v řádcích č. 12 a 16), pak změníte-

li pozici *uipanelu*, automaticky se posunou i vnořená tlačítka. Jejich pozice se nyní počítá v rámci *uipanel* a nikoli v rámci *figure*.

```

3 function Fcn_uipanel
4     F=figure('units','norm','position',[0.1 0.25 0.4 0.38],...
5             'numbertitle','off','color',[0.75 0.75 0.75],...
6             'name','Project ver. 1','tag','figure','menubar','none');
7
8     %--- uipanel ---
9     UIP=uipanel('units','normalized','position',[0.1 0.2 0.20 0.45],...
10               'tag','panel','backgroundcolor',get(gcf,'color'),...
11               'fontsize',16,'fontweight','normal','foregroundcolor','k',...
12               'title','Panel');
13
14     T1=uicontrol(UIP,'units','norm','style','push','string','Barva');
15     set(T1,'position',[0.1 0.55 0.8 0.35],'tag','tlac1','fontsize',14)
16     set(T1,'foregroundcolor',[0 0 0],'backgroundcolor',[0.8 0.8 0.8])
17     set(T1,'callback','set(gcf,'color',[1 0 0])) % udalost tlacka 1
18
19     T2=uicontrol(UIP,'units','norm','style','push','string','Konec');
20     set(T2,'position',[0.1 0.1 0.8 0.35],'tag','tlac2','fontsize',14)
21     set(T2,'foregroundcolor',[0 0 0],'backgroundcolor',[0.8 0.8 0.8])
22     set(T2,'callback','close all') % udalost tlacka 2
23
24     %---
25     %get(UIP)
26     set(UIP,'bordertype')
27     set(UIP,'titleposition')
28     get(UIP,'borderwidth')
29 end

```



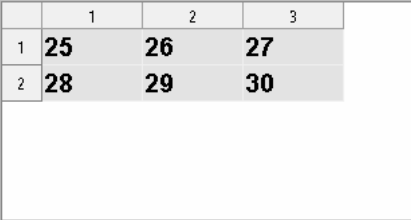
Obr.2.28. Grafický objekt „uipanel“.

Na obr.2.29 je tvorba grafického objektu *uitable*. Je to tabulka, jejíž obsah plní proměnná *d* typu *cell*.

```

1      % dalsi graficke objekty uicontrol
2      %-----
3      function Fcn_uitable
4      -   F=figure('units','norm','position',[0.1 0.25 0.4 0.38],...
5          'numbertitle','off','color',[0.75 0.75 0.75],...
6          'name','Project ver. 1','tag','figure','menubar','none');
7          %--- uitable ---
8      -   d={'25','26','27';'28','29','30'};
9      -   UIT=uitable('units','normalized','position',[0.1 0.2 0.4 0.4],...
10         'tag','table','backgroundcolor',[1 1 0],'data',d,...
11         'fontsize',14,'fontweight','bold');
12         %get (UIT)
13 -   d=get(UIT,'data')|
14 -   whos d
15 -   end

```



	1	2	3
1	25	26	27
2	28	29	30

Obr.2.29. Grafický objekt „uitable“.

3 Tvorba GUI aplikací *App Designer*

Dosud jsme se v této knize zabývali tvorbou aplikací s GUI ručním způsobem pomocí příkazů pro tvorbu grafických objektů. Již delší dobu však byla (a dosud je) v MATLAB vestavěna interaktivní aplikace s názvem *GUIDE*. Spouští se intuitivním příkazem *guide* anebo volbou *New/App/GUIDE* v záložce *HOME*. Uživateli je nabídnuto prázdné okno, do něhož myši přetahuje nabízené grafické objekty, umisťuje je do otevřené plochy, mění jejich velikost a poklepáním myši na grafický objekt nastavuje případné parametry v okně *Inspector*.

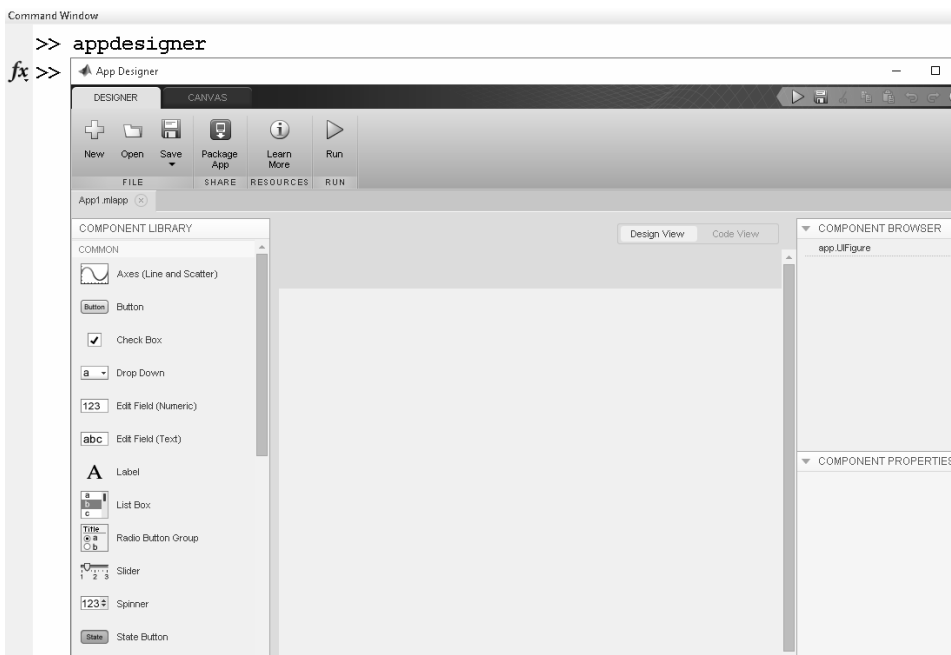
Od verze R2016a se však objevil interaktivní nástroj nový s názvem *App Designer*. Podle tvrzení tvůrce systému MATLAB by měl postupně nahradit starší aplikaci *GUIDE*, která by měla být z MATLAB časem odstraněna. Princip je podobný; přetahování grafických objektů z nabídky do nového okna a poklepáním myši nastavovat jejich parametry. Tato nová aplikace akceptuje moderní trendy v programování, jako jsou např. robustnost, sdílení, apod. Dokonce existuje možnost transformovat projekty s GUI z této starší *GUIDE* do nového *App Designer* (*GUIDE to App Designer Migration Tool for MATLAB*).

Ačkoli popsáný nový přístup umožňuje ruční programování GUI, je *App Designer* primárně určen pro interaktivní práci. Metodou vyber grafický objekt, přetáhni jej do nového okna a nastav parametry plně vystihuje současné trendy i v jiných programovacích prostředích.

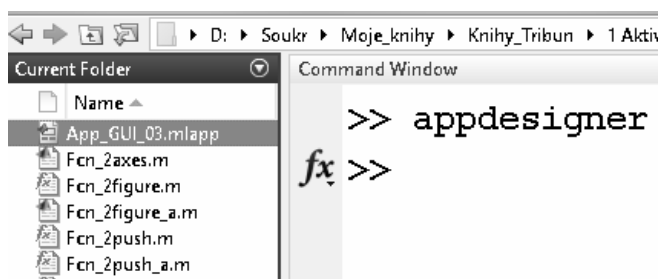
Já osobně pracuji v *App Designer* méně a většinou proto, abych si udržel přehled o nových technologiích a přístupech v tvorbě GUI. Z toho důvodu si v něm dosud nejsem plně jistý v „kramflecích“, byť několik projektů za sebou již mám a tím i jistý přehled. Proto zde uvedu základní přístupy s tím, že pokud vás tento přístup zaujme, hledejte další informace (internet) v podobě textů a videí. Práce v *APP Designer* je toho typu, kdy se věci poměrně snadno provádějí, ale obtížněji popisují.

3.1 Spuštění prostředí a grafický návrh aplikace

Prostředí *App Designer* spustíte buďto z příkazové řádky příkazem *appdesigner* anebo volbou *New/App/App Designer* nahoře vlevo na záložce *HOME*. Prostředí po spuštění ukazuje *obr.3.1*. Roztáhněte jej na celou obrazovku a myši si můžete upravit velikost vnitřního bílého okna. Zatím prázdné okno uložte (*Save*) pod svým jménem, např. *App_GUI_03*. Systém přidá příponu **.mlapp*. Nový soubor se zobrazí v okně *Current Folder*, kde máte nastavenou vaši složku se soubory, viz *obr.3.2*.

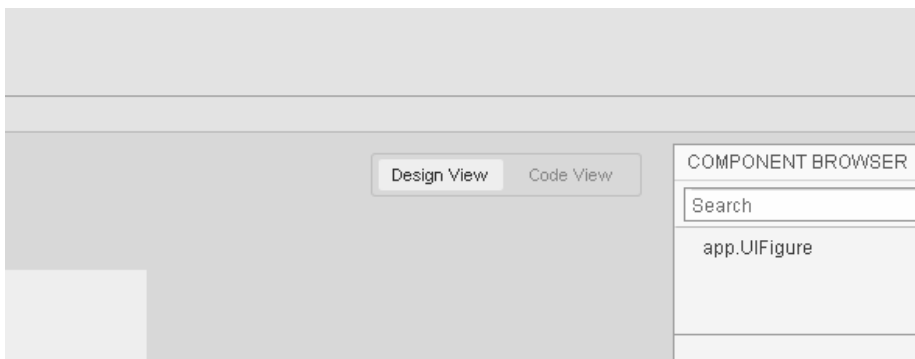


Obr.3.1. Prostor aplikace „APP Designer“ po spuštění.



Obr.3.2. Soubor aplikace „App_GUI_03.mlapp“ ve vaší nastavené složce.

Podívejte se vpravo nad centrálním bílým zatím prázdným oknem. Aplikace nabízí dva základní režimy práce: návrh grafických objektů (*Design View*) a zdrojový kód (*Code View*). Budeme používat oba dva. Vpravo vedle těchto dvou nabídek se objeví název nového prázdného okna: *app.UIFigure*. Můžete jej poklepáním myši změnit na své jméno, ale já to z pedagogických důvodů ponechám na systému. Je to vlastně nové okno, kterému jsme výše říkali grafický objekt *figure*, viz obr.3.3.



Obr.3.3. Dva režimy práce v „App Designer“: grafický návrh (Design View) a zdrojový kód (Code View) spolu s názvem hlavního okna.

Nyní jsme připraveni zahájit přetahování grafických objektů z levé nabídky do bílého okna *app.UIFigure* uprostřed. Na obr.3.4 vidíte stav, kdy jsem z levého svislého panelu vybral potřebné grafické objekty a myší je přetáhl do okna uprostřed. Vybral jsem tyto objekty:

Button (2x) - tlačítka pro *Internet* a *Konec*

Axes (1x) - hlavní kreslicí plátno

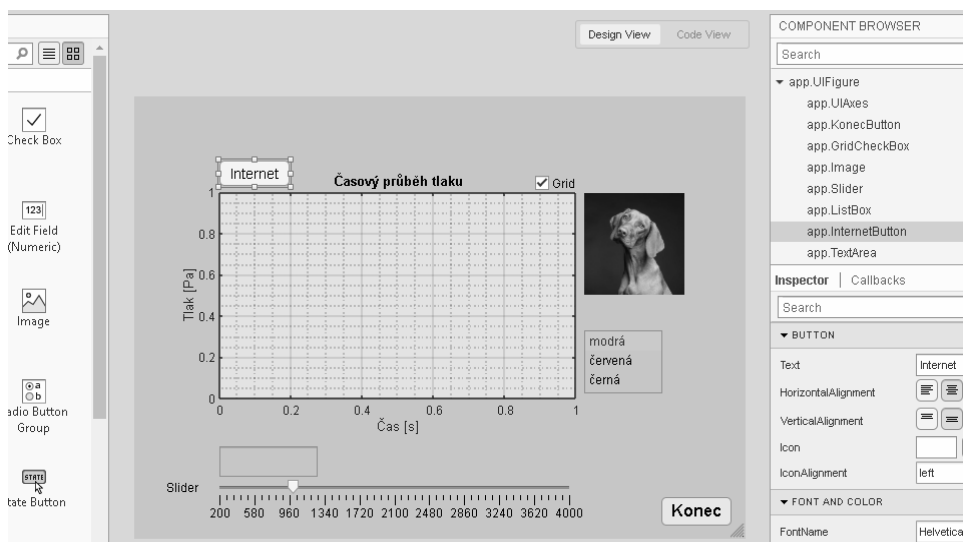
Slider (1x) - tahový potenciometr pro změnu frekvence sinusovky

Check Box (1x) - aktivace a deaktivace mřížky grafu

List Box (1x) - výběr ze tří barev křivky

Image (1x) - vložení obrázku psa

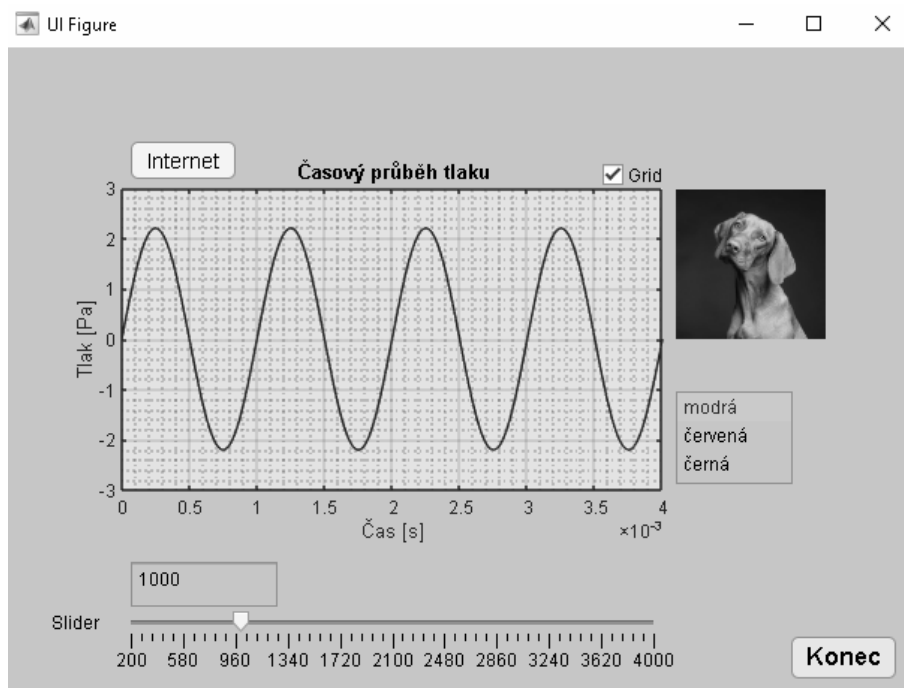
Text Area (1x) - pole pro výpis frekvence nad potenciometrem



Obr.3.4. Přetažení grafických objektů z nabídky vlevo do okna uprostřed.

Všechny vyjmenované grafické objekty najdete v levém svislém okně. Myši je přetáhněte do okna ve středu a můžete upravit jejich polohu a velikost. Všimněte si na *obr.3.4*, že jsem myši vybral tlačítko *Internet*. Jakmile to uděláte, vpravo dole se zobrazí nabídka s možnostmi editace tlačítka (text, zarovnání textu, barva, font, apod.). Současně s tím se zvýrazní vpravo nahoře grafický objekt *app.InternetButton*, což je toto tlačítko. Systém jim přiřazuje automatické názvy, pokud to poklepáním nezměníte. Vidíte zde strom všech grafických objektů, počínaje hlavním oknem *app.UIFigure*. Podle názvů zde uvedených grafických objektů se budeme orientovat při další práci.

Takže máme na ploše přetažené potřebné grafické objekty a nastavené jejich základní parametry. Pohrajte si s tím; vyberte myši grafický objekt, zkoušejte měnit barvy, font, popisy os, titulek grafu, apod. Práci si samozřejmě průběžně ukládejte. Pokud se rozhodnete aplikaci spustit, k tomuto účelu slouží zelený trojúhelník v horní části s názvem *Run* anebo klávesa *F5*, viz podoba aplikace na *obr.3.5*.



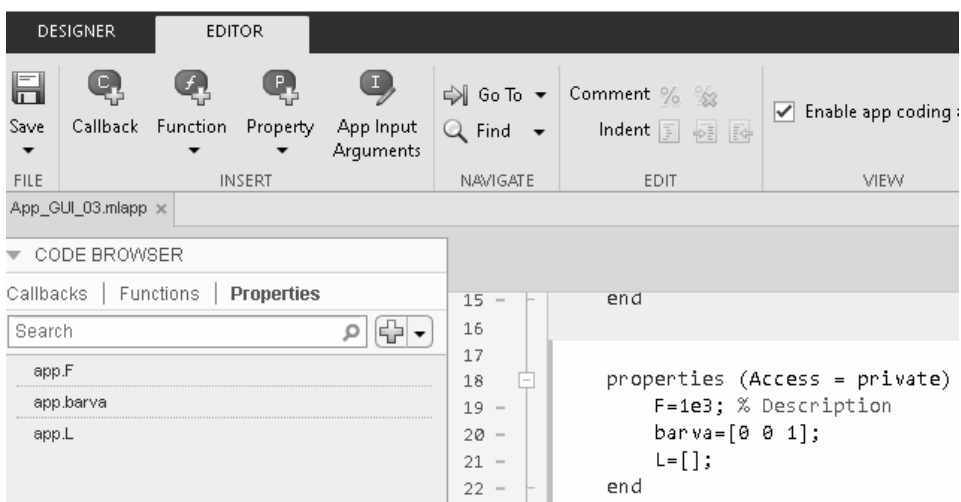
Obr.3.5. Podoba naší aplikace po spuštění

Aplikace se spustí, ale samozřejmě zatím nic nedělá. Naším úkolem nyní bude naprogramovat události a další části kódu.

3.2 Programování událostí a dalších funkcionalit

Začneme třemi věcmi. Tou první bude výběr proměnných, které chceme sdílet ve všech částech naší aplikace. Proměnné můžeme sdílet pomocí vstupních a výstupních parametrů funkcí, ale mimoto je v *App Designer* skvělá možnost vytvořit si jakýsi pomyslný šuplík, do kterého budeme mít přístup odkudkoli a v němž budou námi vybrané proměnné.

Sdílení proměnných v celé aplikaci díky nabídce *Property*



Obr.3.6. Nabídka „Property“ pro sdílení proměnných v celé aplikaci.

Webové stránky v ČR a SR

<https://www.humusoft.cz/> - stránky prodejce systému MATLAB

<https://www.humusoft.cz/events/www-seminars/> - online semináře

<https://www.humusoft.cz/matlab/products/> - aplikační knihovny

<https://www.humusoft.cz/matlab/licensing/> - licence MATLAB

<http://www.matlab.sk> - fórum na téma MATLAB

Webové stránky tvůrce systému MATLAB

<https://www.mathworks.com/> - stránky tvůrce MATLAB v USA

<https://www.mathworks.com/matlabcentral/> - zkušenosti, diskuse

MATLAB®

Grafické uživatelské rozhraní

prof. Ing. Karel Zaplatílek, Ph.D.