

30. Matlab - GUI 1. začíname

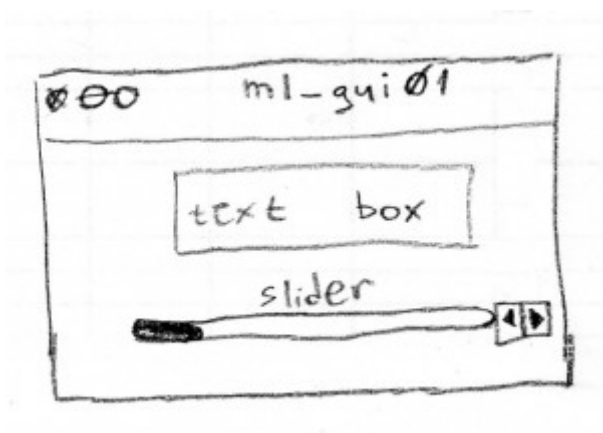
Foltin Martin · Informačné technológie, MATLAB/Comsol

04.03.2011



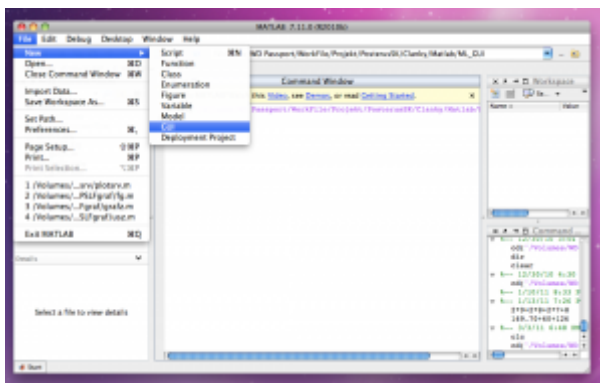
Dnes otvárame ďalšiu rozsiahlu kapitolu prostredia Matlab. Jedná sa o grafické používateľské rozhranie - GUI (graphical user interface). Kapitola bude rozdelená do niekoľko častí, keďže sa jedná o problematiku náročnú a do značnej miery aj rozsiahlu. Budeme sa snažiť postupovať po malých krokoch a radšej niektoré časti zopakujeme, s cieľom správneho pochopenia.

V dnešnej časti si ukážeme, ako GUI vytvoriť, ako doň pridať aktívne prvky a ako ich programovo prepojiť. Bude sa jednať o jednoduchý program, kde bude prepojené textové pole s posuvníkom (slider). Okno aplikácie by teda mohlo vyzeráť asi takto (obr. 1).



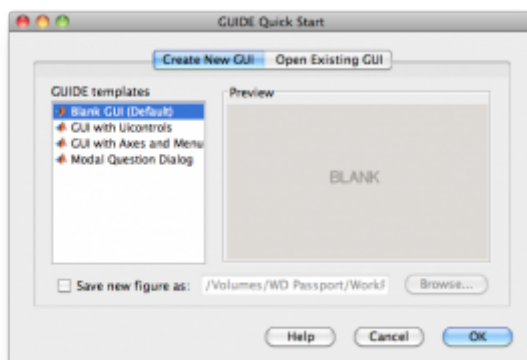
Obr. 1 Grafický návrh aplikácie

Zapneme teda Matlab a spustíme vývojové prostredie pre GUI. Z menu File zvolíme položku New a GUI.



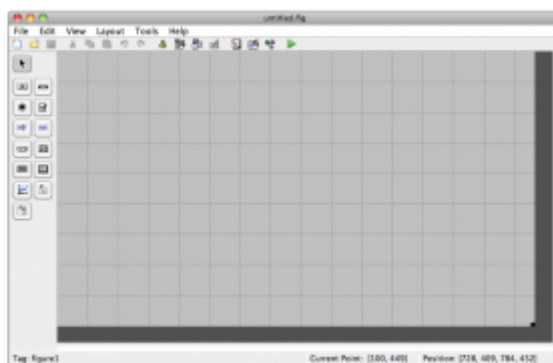
Obr. 2 Spustenie vývojárskeho prostredia pre GUI

Zo sprievodcu zvolíme položku Blank GUI (default).



Obr. 3 GUIDE Quick Start

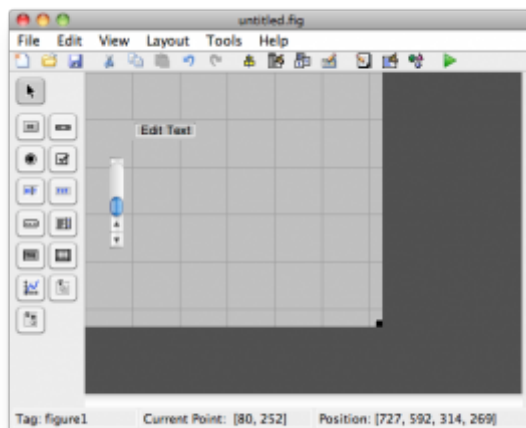
Po tejto voľbe sa spustí prostredie pre vývoj GUI. V ľavej časti sú sústredené jednotlivé aktívne prvky (Push Button, Slider ...), ktoré môžeme využiť v našich aplikáciách. Na pracovnú plochu, ktorá tvorí najväčšiu plochu aplikácie, dostávame aktívne prvky systémom Drag&Drop.



Obr. 4. Prostredie pre tvorbu GUI

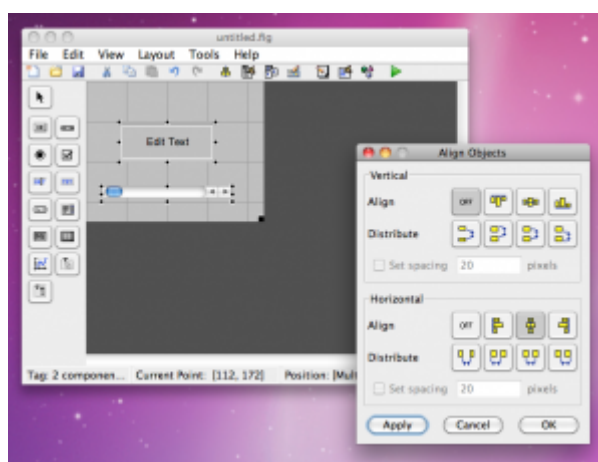
Množina aktívnych prvkov ktoré máme k dispozícii závisí od použitého operačného systému. V našom prípade sa jedná o systém Mac OS 10.6. V operačnom systéme Windows sa nachádza naviac prvok ActiveX. Pomocou neho môžeme dostať do našej aplikácie akýkoľvek prvok typu ActiveX, ktorý ponúka operačný systém Windows.

Umiestnime teda na naše pracovné plátno dva aktívne prvky s ktorými budeme pracovať. Najskôr Edit Text a potom Slider.



Obr. 5. Slider a Edit Text na pracovnej ploche

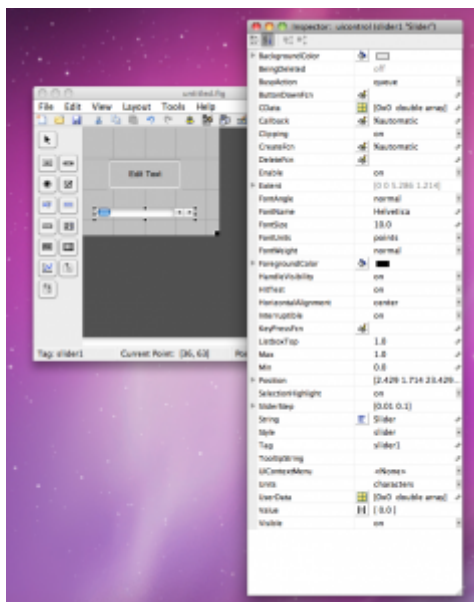
Upravme ich rozmer a zarovnajme ich pomocou nástroja Align Objects. Menu Align Objects vyvoláme kliknutím na ikonu v hornej lište.



Obr. 6 Úprava a zarovnanie objektov

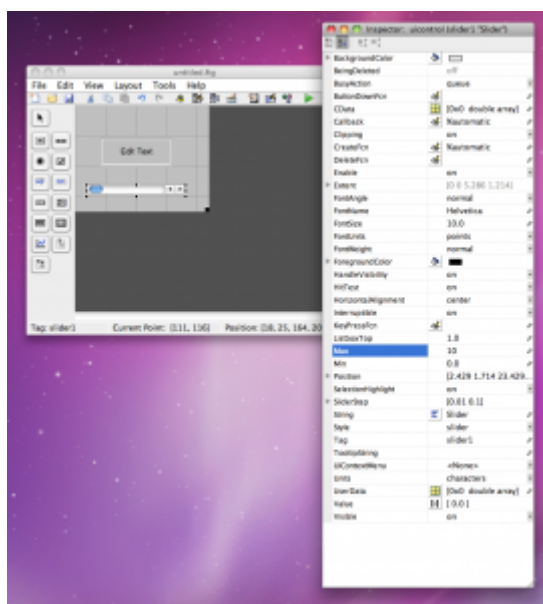
V tomto momente je vhodné uložiť rozpracovanú úlohu. Tento úkon vykonáme voľbou File a Save. Druhá možnosť je využiť ikonu v hornej lište. Zvolíme názov našej aplikácie a adresár kam sa uloží. Matlab sa postará o uloženie súboru typu .fig, kde sú uložené grafické objekty, ich rozmer a poloha v rámci aplikácie. Funkčná stránka aplikácie sa však ukladá do druhého súboru. Ten má príponu .m a nachádzajú sa v ňom všetky funkcie, ktoré sa starajú o správnu prácu aktívnych prvkov. Súboru .m sa budeme venovať v nasledujúcich odstavcoch.

Aktívne prvky majú množinu vlastností ako poloha, farba, popis, názov (Tag), rozsah, prednastavená hodnota a podobne. Pre nastavenie týchto vlastností slúži Property Inspector. Jeho okno vyvoláme dvojklikom na príslušný aktívny prvok.



Obr. 7. Property Inspector

Dôležitá vlastnosť každého použitého objektu je Tag. Jedná sa o jedinečný názov zvoleného aktívneho prvku. Pod týmto názvom sa automaticky generuje kód v .m súbore. Preto voľte toto meno pozorne a venujte mu patričnú pozornosť. Vhodná voľba môže výrazne sprehľadniť program najmä v rozsiahlejších aplikáciách. V našom prípade necháme názvy prednastavené, nakoľko je nami tvorená aplikácia pomerne jednoduchá. Vlastnosť Tag je teda v prípade objektu slider slider1 a v prípade Edit Text edit1. Vlastnosť, ktorú je možné v našej aplikácii zmeniť je napr. Rozsah posuvníka. Nech minimum je 0 a maximum 10. Postačuje iba zmeniť príslušnú vlastnosť

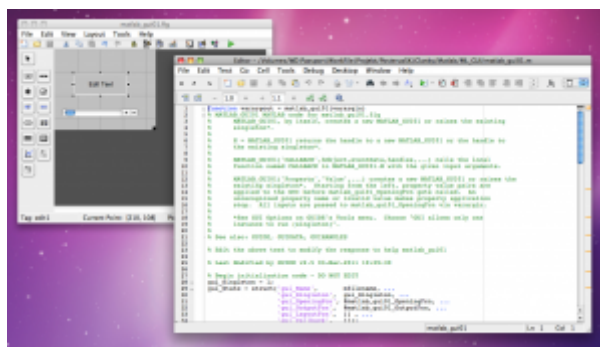


Obr. 8. Vlastnosť Max objektu slider1

Podobným spôsobom môžeme zmeniť nápis, ktorý sa zobrazuje v okne Edit Text. Tentokrát sa jedná o zmenu vlastnosti String. Nastavme ju na hodnotu NULL. V tomto okamihu môžeme považovať návrh grafickej časti aplikácie za ukončený. Preto je vhodné pred ďalšou prácou uložiť ju.

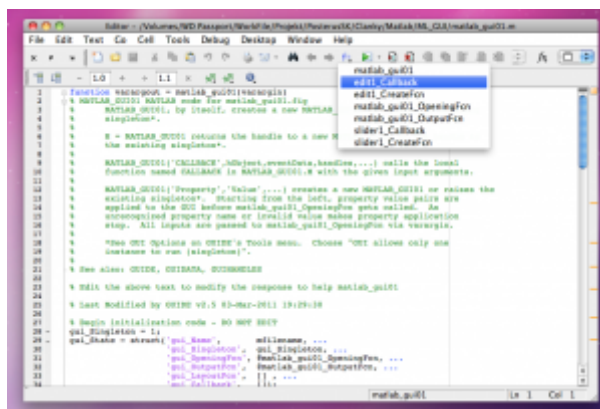
Pokračovať budeme tvorbou programu, ktorý prepojí posuvník s textovým oknom.

Cieľom je dosiahnuť, že pri zmene polohy posuvníka sa automaticky prepíše hodnota v textovom okne a pri prepise hodnoty v textovom okne sa posunie slider na pozíciu zodpovedajúce napísanej hodnote. Všetky tieto úkony musíme vykonať programovo v automaticky vygenerovanom .m súbore. M súbor vyvoláme kliknutím na ikonu M-file Editor v hornej lište.



Obr. 9 M-file Editor

Pri pohľade na zdrojový kód vidíme, že Matlab automaticky vygeneroval niekoľko funkcií. Medzi najdôležitejšiu patrí funkcia s príponou `_OpeningFcn`. Jedná sa o funkciu, ktorá sa vykoná počas prvého spustenia aplikácie. Služi na nastavenie štartovacích hodnôt využívaných aplikáciou. Podrobnejšie sa jej budeme venovať v budúcej časti. My sa dnes sústreďíme na funkcie s príponou `_Callback`. Takúto funkciu Matlab vygeneruje pre každý aktívny prvok, ktorý sme umiestnili na pracovnú plochu aplikácie. Prefix funkcie tvorí práve názov aktívneho prvku vo vlastnosti `Tag`. V našom prípade budú k dispozícii funkcie `edit1_Callback` a `slider1_Callback`.



Obr. 10 Zoznam funkcií v rámci .m súboru

Nakoľko je každá akcia previazaná s aktívnym prvkom zapúzdrená vo funkcii je potrebné vysvetliť spôsob odovzdávania dát medzi jednotlivými funkciami. Napreč celým GUI je prepletená dátová štruktúra s menom handles. K tejto špeciálnej premennej je možné pristupovať v každej funkcii. Spôsob ako odovzdávať dáta pomocou špeciálnej premennej si ukážeme v budúcej časti. Dnes sa sústreďíme na spôsob získavania vlastností z aktívnych prvkov. Štruktúra handles hrá v tomto procese nezastupiteľné miesto. Ku každej vlastnosti a k jej hodnote je možné pristupovať takouto konštrukciou

```
handles.Tag_objektu(nazov_vlastnosti)
```

Takáto konštrukcia sa dá použiť ako pri čítaní vlastnosti, ale aj pri jej zmene. K tomuto

účelu slúži dvojica príkazov get a set. Ak teda potrebujeme zistiť hodnotu, ktorá sa nachádza na posuvníku, použijeme takýto zápis

```
get(handles.slider1('Value'))
```

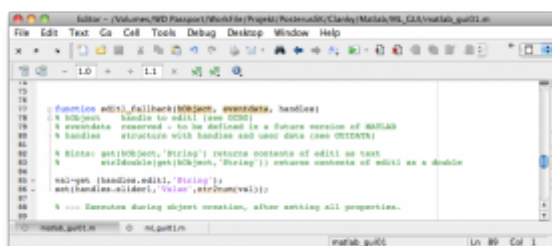
Analogicky by sme postupovali ak by sme chceli zmeniť hodnotu posuvníka.

```
set(handles.slider1,'Value',5)
```

V tomto momente už disponujeme potrebnými znalosťami na to, aby sme dokázali našu aplikáciu dokončiť. Zapišme teda potrebný kód do príslušných Callbacks. Najskôr nech sú to riadky pre edit1_Callback.

```
val=get (handles.edit1,'String');
set(handles.slider1,'Value',str2num(val));
```

V prvom riadku do premennej val ukladáme reťazec, ktorý je zapísaný v editovacom okne typu Edit Text. Jedná sa o hodnotu typu string. Rovnako je aj nazvaná vlastnosť ktorá uchováva zadaný text do okna. V druhom riadku pomocou príkazu set zapisujeme do posuvníka slider1 hodnotu prečítanú z textového okna. Všimnite si, že musí dôjsť k pretypovaniu. Vlastnosť objektu slider1 Value očakáva numerickú a nie reťazcovú hodnotu. Preto je potrebné použiť príkaz str2num.

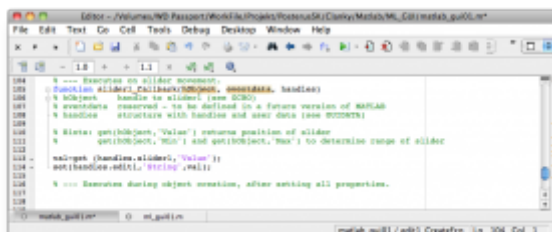


Obr. 11 edit1_Callback

Obdobným spôsobom zapišeme kód aj do slider1_Callback.

```
val=get (handles.slider1,'Value');
set(handles.edit1,'String',val);
```

V prvom riadku prečítame hodnotu posuvníka slider1 a uložíme do numerickej premennej val. V druhom riadku zapišeme hodnotu do textového okna príkazom set. Všimnite si, že v tomto prípade meníme hodnotu vlastnosti String. V tomto prípade nie je potrebné pretypovanie.



Obr. 12 slider1_Callback

V tomto momente je aplikácia hotová. Postačuje je už iba spustiť. Môžete tak spraviť priamo z príkazového riadku Matlabu, z M-file Editora, alebo z grafického editora. Výsledkom je jednoduchá aplikácia v ktorej komunikuje posuvník s oknom typu Edit Text a obrátene.



Obr. 13 hotové GUI



Video Tvorba jednoduchého GUI

Táto aplikácia nech je pre vás úvodom pre tvorbu náročnejších programov. V budúcej časti sa naučíme tvoriť zložitejšie rozhrania, v ktorých budeme využívať už aj grafiku, tlačidlá a roletové menu. Taktiež si priblížime význam funkcie `_OpeningFcn`.