



TECHNICKÁ UNIVERZITA V LIBERCI
**Fakulta mechatroniky, informatiky
a mezioborových studií**



MATLB: přednáška 6

Tvorba grafických uživatelských rozhraní

Jaroslav Čmejla

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- **Kdo bude uživatel?**
- K čemu má GUI sloužit?
- Jak bude vypadat?
- Tvorba GUI v Matlabu
 - Ruční programování pomocí příkazu `uicontrol`
 - GUIDE (zastaralý a již nebude podporován)
 - App Designer

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- Kdo bude uživatel?
- **K čemu má GUI sloužit?**
- Jak bude vypadat?
- Tvorba GUI v Matlabu
 - Ruční programování pomocí příkazu `uicontrol`
 - GUIDE (zastaralý a již nebude podporován)
 - App Designer

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- Kdo bude uživatel?
- K čemu má GUI sloužit?
- **Jak bude vypadat?**
- Tvorba GUI v Matlabu
 - Ruční programování pomocí příkazu `uicontrol`
 - GUIDE (zastaralý a již nebude podporován)
 - App Designer

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- Kdo bude uživatel?
- K čemu má GUI sloužit?
- Jak bude vypadat?
- **Tvorba GUI v Matlabu**
 - Ruční programování pomocí příkazu `uicontrol`
 - GUIDE (zastaralý a již nebude podporován)
 - App Designer

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- Kdo bude uživatel?
- K čemu má GUI sloužit?
- Jak bude vypadat?
- Tvorba GUI v Matlabu
 - **Ruční programování pomocí příkazu `uicontrol`**
 - GUIDE (zastaralý a již nebude podporován)
 - App Designer

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- Kdo bude uživatel?
- K čemu má GUI sloužit?
- Jak bude vypadat?
- Tvorba GUI v Matlabu
 - Ruční programování pomocí příkazu `uicontrol`
 - **GUIDE (zastaralý a již nebude podporován)**
 - App Designer

Tvorba grafických uživatelských rozhraní (GUI) v Matlabu

Postup programování

- Kdo bude uživatel?
- K čemu má GUI sloužit?
- Jak bude vypadat?
- Tvorba GUI v Matlabu
 - Ruční programování pomocí příkazu `uicontrol`
 - GUIDE (zastaralý a již nebude podporován)
 - **App Designer**

Co je nová aplikace v Matlabu

■ GUI je figure

- Aplikace čeká na akci uživatele
- Každý objekt (figure, tlačítka, radioboxy, ...) má své vlastnosti, které lze načíst a měnit příkazy `get` a `set`
- Každý objekt má `handle`, což je obdoba ukazatele
- `Callback` je vlastnost, ve které se nastavuje ukazatel na funkci, která se zavolá při dané události (stisk tlačítka, průchod myši, ...)
- Program je řízený událostmi, které generuje Matlab (a uživatel).

Co je nová aplikace v Matlabu

- GUI je figure
- **Aplikace čeká na akci uživatele**
- Každý objekt (figure, tlačítka, radioboxy, ...) má své vlastnosti, které lze načíst a měnit příkazy `get` a `set`
- Každý objekt má `handle`, což je obdoba ukazatele
- `Callback` je vlastnost, ve které se nastavuje ukazatel na funkci, která se zavolá při dané události (stisk tlačítka, průchod myši, ...)
- Program je řízený událostmi, které generuje Matlab (a uživatel).

Co je nová aplikace v Matlabu

- GUI je figure
- Aplikace čeká na akci uživatele
- Každý objekt (figure, tlačítka, radioboxy, ...) má své vlastnosti, které lze načíst a měnit příkazy `get` a `set`
- Každý objekt má `handle`, což je obdoba ukazatele
- `Callback` je vlastnost, ve které se nastavuje ukazatel na funkci, která se zavolá při dané události (stisk tlačítka, průchod myši, ...)
- Program je řízený událostmi, které generuje Matlab (a uživatel).

Co je nová aplikace v Matlabu

- GUI je figure
- Aplikace čeká na akci uživatele
- Každý objekt (figure, tlačítka, radioboxy, ...) má své vlastnosti, které lze načíst a měnit příkazy `get` a `set`
- Každý objekt má `handle`, což je obdoba ukazatele
- Callback je vlastnost, ve které se nastavuje ukazatel na funkci, která se zavolá při dané události (stisk tlačítka, průchod myši, ...)
- Program je řízený událostmi, které generuje Matlab (a uživatel).

Co je nová aplikace v Matlabu

- GUI je figure
- Aplikace čeká na akci uživatele
- Každý objekt (figure, tlačítka, radioboxy, ...) má své vlastnosti, které lze načíst a měnit příkazy `get` a `set`
- Každý objekt má `handle`, což je obdoba ukazatele
- **Callback je vlastnost, ve které se nastavuje ukazatel na funkci, která se zavolá při dané události (stisk tlačítka, průchod myši, ...)**
- Program je řízený událostmi, které generuje Matlab (a uživatel).

Co je nová aplikace v Matlabu

- GUI je figure
- Aplikace čeká na akci uživatele
- Každý objekt (figure, tlačítka, radioboxy, ...) má své vlastnosti, které lze načíst a měnit příkazy `get` a `set`
- Každý objekt má `handle`, což je obdoba ukazatele
- `Callback` je vlastnost, ve které se nastavuje ukazatel na funkci, která se zavolá při dané události (stisk tlačítka, průchod myši, ...)
- Program je řízený událostmi, které generuje Matlab (a uživatel).

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: gcf, gca, get, set
- Na ukládání vlastních dat aplikace používáme vlastnost objektů UserData, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do UserData jednoho objektu (např. figure), nebo bude používat více objektů.
- UserData tedy plní funkci datového segmentu aplikace

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: gcf, gca, get, set
- Na ukládání vlastních dat aplikace používáme vlastnost objektů UserData, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do UserData jednoho objektu (např. figure), nebo bude používat více objektů.
- UserData tedy plní funkci datového segmentu aplikace

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: gcf, gca, get, set
- Na ukládání vlastních dat aplikace používáme vlastnost objektů UserData, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do UserData jednoho objektu (např. figure), nebo bude používat více objektů.
- UserData tedy plní funkci datového segmentu aplikace

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: `gcf`, `gca`, `get`, `set`
- Na ukládání vlastních dat aplikace používáme vlastnost objektů `UserData`, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do `UserData` jednoho objektu (např. figure), nebo bude používat více objektů.
- `UserData` tedy plní funkci datového segmentu aplikace

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: gcf, gca, get, set
- Na ukládání vlastních dat aplikace používáme vlastnost objektů UserData, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do UserData jednoho objektu (např. figure), nebo bude používat více objektů.
- UserData tedy plní funkci datového segmentu aplikace

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: gcf, gca, get, set
- Na ukládání vlastních dat aplikace používáme vlastnost objektů UserData, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do UserData jednoho objektu (např. figure), nebo bude používat více objektů.

■ UserData tedy plní funkci datového segmentu aplikace

Programování pomocí GUIDE

- Objekty vkládáme v GUIDE, nastavujeme vlastnosti, callbacky
- Funkce volané callbacky mají pevnou strukturu vstupních parametrů
- Ukládáme do souborů *.m (funkce - callbacky) a *.fig (aplikace)
- Často používané příkazy: gcf, gca, get, set
- Na ukládání vlastních dat aplikace používáme vlastnost objektů UserData, která může obsahovat jakýkoliv datový typ (čili všechno).
- Je na programátorovi, jestli bude ukládat všechna data do UserData jednoho objektu (např. figure), nebo bude používat více objektů.
- **UserData tedy plní funkci datového segmentu aplikace**

“Ruční” programování

- Výhoda: nepotřebujeme soubor *.fig, celá aplikace je v jednom (nebo více) m-filu (ještě spolehlivější přenositelnost na různé verze Matlabu)
- Objekty vkládáme pomocí příkazu `uicontrol`, např. tlačítko

```
uicontrol( ...  
    'Style','push', ... % styl nastavuje typ objektu  
    'Units','normalized', ...  
    'Position',[0.80 0.94 0.16 0.05], ...  
    'String','KONEC', ...  
    'FontWeight','bold',...  
    'ForegroundColor',[1 0 0],...  
    'Callback','funkcecallbacku');
```

“Ruční” programování

- Výhoda: nepotřebujeme soubor *.fig, celá aplikace je v jednom (nebo více) m-filu (ještě spolehlivější přenositelnost na různé verze Matlabu)
- Objekty vkládáme pomocí příkazu `uicontrol`, např. tlačítko

```
uicontrol( ...  
    'Style','push', ... % styl nastavuje typ objektu  
    'Units','normalized', ...  
    'Position',[0.80 0.94 0.16 0.05], ...  
    'String','KONEC', ...  
    'FontWeight','bold',...  
    'ForegroundColor',[1 0 0],...  
    'Callback','funkcecallbacku');
```

“Ruční” programování

- Událost můžeme odchytit samostatnou funkcí nebo jednou funkcí s parametrem.
- Je-li aplikace ovládána jednou funkcí, odchytáváme událost většinou pomocí `switch`.

“Ruční” programování

- Událost můžeme odchytit samostatnou funkcí nebo jednou funkcí s parametrem.
- Je-li aplikace ovládána jednou funkcí, odchytáváme událost většinou pomocí `switch`.

Struktura typické aplikace řízené jednou funkcí

```
function mojeaplikace(udalost)
```

```
if nargin==0, udalost='zacatek'; end
```

```
switch udalost
```

```
    case 'zacatek'
```

```
        fig=figure(...
```

```
        data.tlacitko1=uicontrol('Style','push',...
```

```
        ...
```

```
        'Callback','mojeaplikace(''stisktlacitka1'')');
```

```
        ...
```

```
        set(fig,'UserData',data); % uložení vlastních dat
```

```
    case 'stisktlacitka1'
```

```
        data=get(gcf,'UserData');
```

```
        ...
```

```
        set(gcf,'UserData',data);
```

```
end % switch
```

App Designer

- **Objektově orientovaný způsob programování**
- Aplikace je vytvářena jako nový (zděděný) objekt s novými vlastnostmi a metodami
- Aplikace vytvořené v GUIDE lze migrovat do AppDesigneru

App Designer

- Objektově orientovaný způsob programování
- Aplikace je vytvářena jako nový (zděděný) objekt s novými vlastnostmi a metodami
- Aplikace vytvořené v GUIDE lze migrovat do AppDesigneru

App Designer

- Objektově orientovaný způsob programování
- Aplikace je vytvářena jako nový (zděděný) objekt s novými vlastnostmi a metodami
- Aplikace vytvořené v GUIDE lze migrovat do AppDesigneru

Příklady

- Demo aplikace makevase
- Nástroj Filter Design fdatool
- Aplikace eegplot

Tento materiál vznikl v rámci projektu ESF CZ.1.07/2.2.00/28.0050

Modernizace didaktických metod a inovace výuky technických předmětů,
který je spolufinancován Evropským sociálním fondem a státním rozpočtem ČR.

Příklady

- Demo aplikace makevase
- **Nástroj Filter Design fdatool**
- Aplikace eegplot

Tento materiál vznikl v rámci projektu ESF CZ.1.07/2.2.00/28.0050

Modernizace didaktických metod a inovace výuky technických předmětů,
který je spolufinancován Evropským sociálním fondem a státním rozpočtem ČR.

Příklady

- Demo aplikace makevase
- Nástroj Filter Design fdatool
- Aplikace eegplot

Tento materiál vznikl v rámci projektu ESF CZ.1.07/2.2.00/28.0050

Modernizace didaktických metod a inovace výuky technických předmětů,
který je spolufinancován Evropským sociálním fondem a státním rozpočtem ČR.