

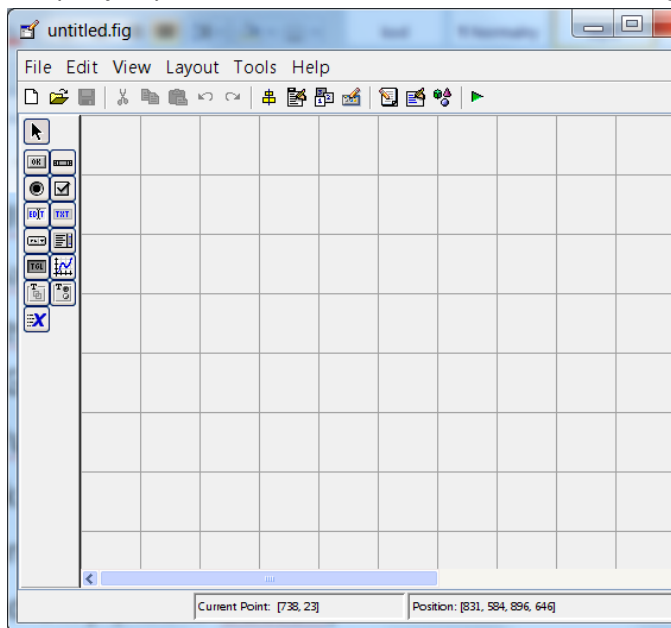
Ćwiczenie 12. Matlab – formularze, komponenty

Tworzenie interfejsu komponentowego GUI (Graphic User Interface)

Wpisujemy w *Command Window*:

guide

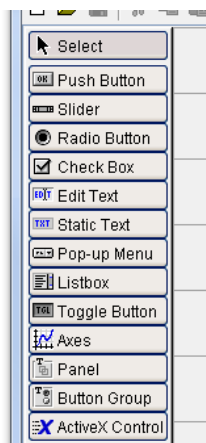
Otrzymujemy okienko z możliwościami tworzenia nowego formularza (lub edycji istniejącego):



Poniżej narzędzia wskazywania elementów po lewej stronie mamy do wyboru następujące komponenty:

- Przycisk (*Push Button*)
- Suwak (*Slider*)
- Przycisk radiowy (*Radio Button*)
- Pole wyboru (*Checkbox*)
- Pole tekstowe (*Edit Text*)
- Etykieta (*Static Text*)
- Lista rozwijana (*Popup Menu*)
- Lista wyboru (*Listbox*)
- Przełącznik (*Toggle Button*)
- Wykres
- Panel
- Grupa przycisków (*Button group*)
- komponent ActiveX

Uwaga: aby widzieć opisy komponentów z menu *File/Preferences* wybrać pozycję *GUIDE* i zaznaczyć opcję *Show names in component palette*.



Projektowanie aplikacji

Tworzymy nową aplikację – *Blank GUI*. Wybieramy myszką potrzebny komponent i klikamy w obszar roboczy (lub rysujemy w oknie).

Po dwukrotnym kliknięciu w dowolne miejsce obszaru okna aplikacji (lub w dowolny utworzony komponent) pojawia się okno *Property Inspector* – właściwości danego komponentu. (Inne możliwości uruchomienia inspektora to: kliknięcie prawym przyciskiem myszy i z rozwijanego menu wybranie opcji *Property Inspector* lub z menu *View* wybranie *Property Inspector*). W oknie *Property Inspector* można ustawić podstawowe informacje o danym komponencie.

Ćwiczenie

1. Ustalmy wielkość okna aplikacji w inspektorze, Klikamy dwukrotnie w obszar okna aplikacji GUI:

- Units - wybieramy *centimeters*
- Position - naciskamy „+” i zmieniamy *width* na 15, a *height* na 10

2. Wstawiamy przycisk *Push Button* (można też wstawić komponent wykresu *Axes*, ale niekoniecznie).

Napisy na przyciskach określa właściwość *String*. Ważna też jest właściwość *Tag* - nazwa komponentu, pod którą jest on identyfikowany w kodzie.

3. Nadajemy w inspektorze dla przycisku *Push Button*:
 właściwość *String* (to jest tekst na przycisku)- wpisujemy *OK*
 właściwość *Tag* (to jest nazwa programowa komponentu) - wpisujemy *Rysuj*
 właściwość *FontSize* – wpisujemy 14
 właściwość *Position* (rozwijana przyciskiem +) – sprawdzić składowe.

4. Dodajemy komponent *Axes*.

5. Zapisujemy aplikację pod nazwą *wykres*

Oprócz pliku *wykres.fig* z projektem formatki, tworzony jest też m-plik o identycznej nazwie i otwierany w edytorze *Matlaba*. Jest tam wiele funkcji dotyczących projektu i jego komponentów, wiele też anglojęzycznych komentarzy (poprzedzonych znakiem %), które można przeczytać albo usunąć.

Uwaga: otwieranie gotowych projektów aplikacji w celu ich modyfikacji z menu kontekstowego zapisanego pliku *fig* - *Open in GUIDE*.

Dodanie funkcjonalności dla aplikacji – funkcje *Callback*

Ćwiczenie

Dodajmy teraz funkcjonalność przyciskowi.

1. Z menu kontekstowego przycisku *rysuj* z napisem *OK* wybieramy *View Callbacks/Callback*. Przenosimy się do funkcji *rysuj_Callback* w *m-pliku* aplikacji.

Funkcja ***Callback*** będzie wykonywana po kliknięciu przycisku uruchomionej aplikacji.

2. Poniżej nagłówka funkcji:

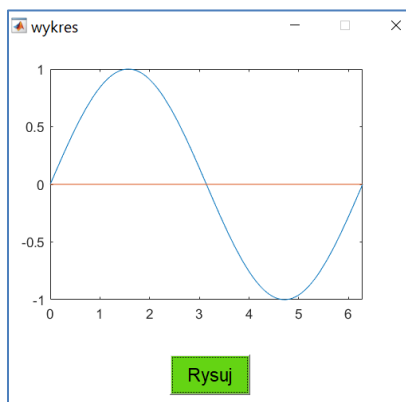
```
function rysuj_Callback(hObject, eventdata, handles)
```

dopisujemy instrukcję rysowania wykresu (symbolicznie):

```
syms x  
fplot ([sin(x) 0],[0 2*pi]);
```

Możemy też przenosić się do kodu funkcji w *m-pliku* korzystając z pozycji *GoTo* w menu edytora *m-pliku* i wybrać właściwą dla danego komponentu funkcję *Callback*.

3. Uruchamiamy program przyciskiem *Run* w oknie *m-pliku* (lub F5 na klawiaturze) albo też w oknie formatki aplikacji. W oknie działającego programu klikamy przycisk, którego *Callback* powinien wykonać nasz wykres sinusoidy.



Sterowanie aplikacją

Dostęp do właściwości komponentów

Po uruchomieniu aplikacji wszystkie komponenty umieszczane są w zmiennej o nazwie ***handles***, która jest strukturą obiektów. Zamiar pobrania (do zmiennej) lub zmiany właściwości dowolnego komponentu realizowany jest z wykorzystaniem dwóch funkcji:

```
zmienna = get (handles.tag_obiektu, właściwość) - pobranie właściwości do zmiennej  
set (handles.tag_obiektu, właściwość, wartość) - ustawienie właściwości
```

Ćwiczenie

1. W nowej aplikacji wstawiamy komponent *Edit Text*.

2. Znajdujemy w kodzie m-pliku funkcję o nazwie *naszplik_OpeningFcn*. Jest ona wykonywana zaraz po uruchomieniu aplikacji. Poniżej instrukcji:

```
guidata(hObject, handles);
```

wpisujemy

```
set(handles.edit1,'String','Tekst po starcie');
```

Sprawdzamy działanie aplikacji.

3. Wstawiamy przycisk *Push Button*. W kodzie jego funkcji Callback wpisujemy:

```
set(handles.edit1,'String','Tekst po kliknięciu');
```

4. Sprawdzamy działanie aplikacji.

Zadanie

Utworzyć aplikację z dwoma przyciskami *Push Button* i dwoma komponentami *Axes*. Kliknięcie przycisków powoduje tworzenie wykresów $\sin(x)$ i $\cos(x)$ w odpowiednim układzie współrzędnych. Uwaga: wybór w którym komponente ma być rysowany wykres uzyskujemy wykonując instrukcję:

```
axes(handles.tag_odpowiedniego_axes);
```

Sterowanie wartością zmiennej przy pomocy suwaka (Slider)

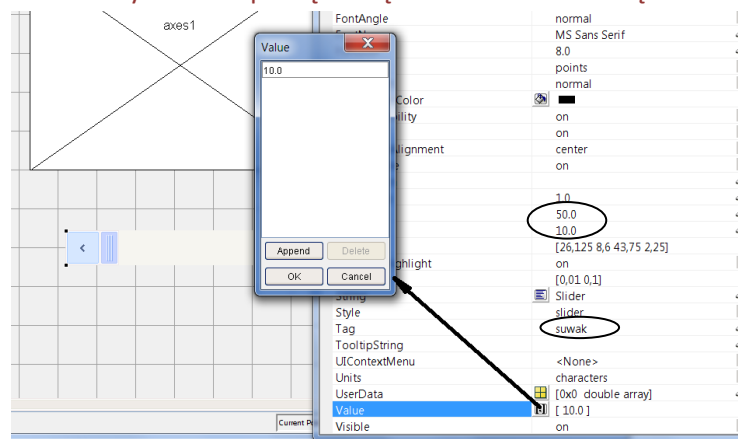
1. Wstawiamy suwak (*Slider*) i ustalamy w inspektorze jego właściwości:

Tag : suwak

Min:10

Max:50

a także w osobnym okienku ustalamy wartość początkową suwaka *Value* równą 10:



2. Dodajemy w kodzie dla funkcji *Callback* suwaka (akcja przy przesuwaniu suwaka):

```
function suwak_Callback(hObject, eventdata, handles)
```

```
pozycja=get(hObject,'Value'); %pobranie Value suwaka (hObject to obiekt tej funkcji Callback)
```

```
%zamiast hObject można pisać handles.suwak
```

```
z=round(pozycja); %zaokrąglenie do liczby całkowitej
```

```
sphere(z); %rysowanie wykresu kuli z argumentem
```

albo

```
syms x
```

```
ezplot(sin(z*x),[0 pi]); %rysowanie funkcji sinus z różnymi częstotliwościami
```

Tak utworzona aplikacja po przesunięciu suwaka ustali (według pozycji suwaka) wartość parametru wykresu dla funkcji. Można również dopisać w kodzie funkcji *plik_OpeningFcn* instrukcję:

```
peaks(10); (lub: fplot('sin(10*x)',[0 2*pi]))
```

aby zaraz po uruchomieniu aplikacji pojawiał się wykres.

Wykorzystanie właściwości obiektu w funkcji innego obiektu.

Ćwiczenie

Celem aplikacji jest rysowanie wykresu po ustawieniu suwaka, ale dopiero w reakcji na kliknięcie przycisku, np. wykres funkcji $\sin(x)$ w przedziale zmiennej x od 0 do wartości ustalonej pozycją suwaka

1. Umieszczamy dodatkowo w oknie przycisk *Push Button*, zmieniamy *String* przycisku na *Kliknij* i *Tag* na *rysuj*.

2. Wpisujemy w kodzie funkcji *rysuj_Callback*:

```
function rysuj_Callback(hObject, eventdata, handles)
```

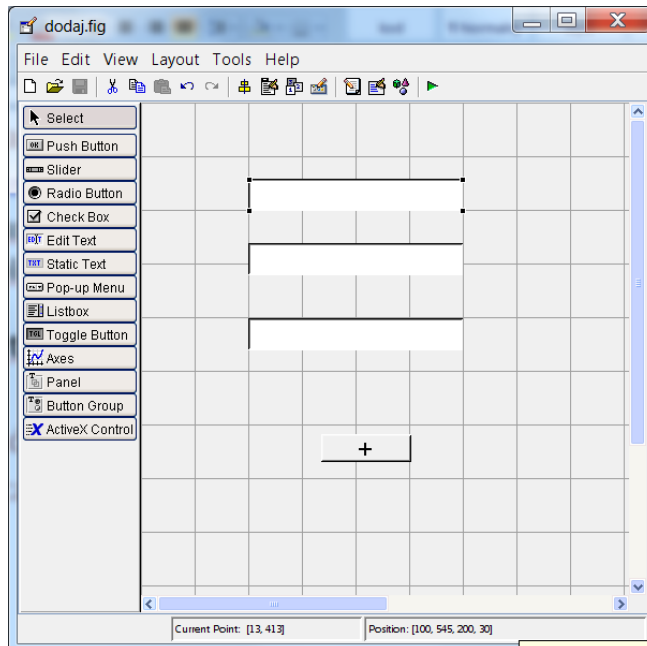
```
z=get(handles.suwak,'Value'); %pobranie wartości Value suwaka
fplot('sin(x)',[0, pi*round(z)]); %rysowanie wykresu z ustalonym argumentem
```

Obliczenia arytmetyczne

Ćwiczenie

Wykonać aplikację do sumowania dwóch liczb dziesiętnych.

- Umieszczamy na formularzu trzy komponenty *Edit Text* i jeden przycisk *Push Button*.



- Korzystając z *Inspektora* - polom edycyjnym zmieniamy *tagi* na e1, e2 i e3 oraz usuwamy napisy we właściwościach *String*. Na przycisku umieszczamy *String* jako znak + i ustalamy *Tag* na *dodaj*.
- W kodzie funkcji *Callback* przycisku *dodaj* piszemy:

```
function klik_Callback(hObject, eventdata, handles)
s1=str2double(get(handles.e1,'String'));
s2=str2double(get(handles.e2,'String'));
suma=s1+s2;
set(handles.e3,'String',num2str(suma));
guidata(hObject, handles);
```

Uwaga: zastosowano powyżej funkcje konwersji typu danych:

str2double(tekst) - konwersja tekstu na liczbę
num2str(liczba) - konwersja liczby na tekst

Zadania

- Do uprzednio utworzonej aplikacji dodać nowe przyciski, których zadaniem będzie:
 - czyszczenie wszystkich pól edycyjnych – przypisanie im pustych ciągów znaków: ''
 - wykonanie operacji mnożenia dwóch liczb,
 - obliczenie średniej arytmetycznej z dwóch liczb,
 - obliczanie pierwiastka stopnia n (liczbę wpisujemy do pierwszego pola *Edit Text*, a rząd pierwiastka do drugiego pola).
- Utworzyć aplikację do obliczania długości tekstu wpisanego do pola *Edit Text* (wykorzystać funkcję *length*).
- Utworzyć aplikację prostego kalkulatora. Wstawiamy komponenty *Edit Text* i 10 przycisków *Push Button* do wpisywania cyfr oraz jeden *Push Button* do wykonania obliczenia.

Dla przycisków z cyframi od 0 do 9 określamy odpowiednie napisy (*String*) w *Inspektorze*. Tworzymy dla nich kody funkcji *Callback* według wzoru:

```
tekst=get(handles.e1,'String');
set(handles.e1,'String',strcat(tekst,'1'));
```

i tak dalej, zmieniając cyfrę dopisywaną. Odbywa się tu pobranie z komponentu *Edit Text* ciągu znaków i dołączenie klikniętej cyfry (łącznie napisów wykonuje funkcja *strcat*).

Callback przycisku kasowania pola *Edit Text* powinien mieć kod:

```
set(handles.e1,'String','');
```

Callback przycisku obliczającego np. funkcję *sinus* ma kod:

```
tekst=get(handles.e1,'String');
```

```
wynik=sind(str2double(tekst)); % sind - kąt będzie w stopniach
```

```
set(handles.e1,'String', num2str(wynik));
```