



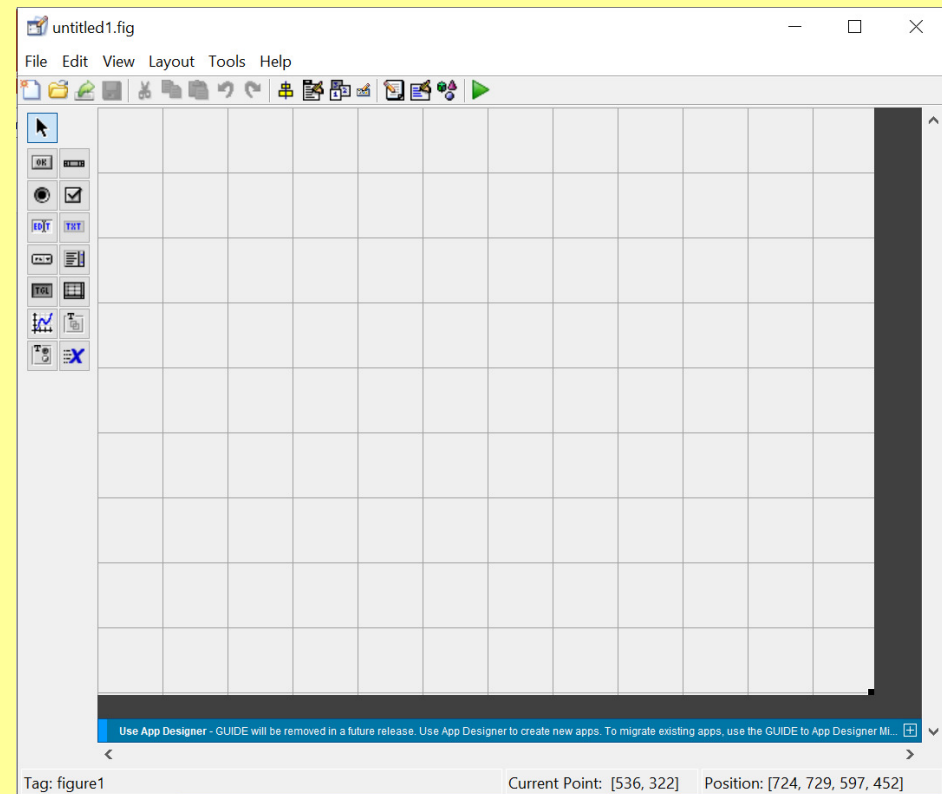
# Wykład 8

## GUI – Graficzny Interfejs Użytkownika

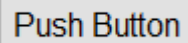
Graficzny interfejs użytkownika w MATLABie tworzymy wpisując w oknie poleceń komendę **guide** (nowa wersja nazywa się **App Designer**). Aby zaprojektować nową nakładkę wybieramy opcję **Blank Gui**. Po zakończeniu pracy powstaje plik \*.fig zawierający rozmieszczenie poszczególnych elementów oraz plik \*.m zawierający kod programu. Możliwa jest konwersja interfejsu utworzonego poleceniem **guide** do nowej wersji (*Migrate to App Designer*).

Po wybraniu Blank Gui pojawia się okno:

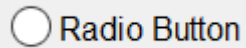
Na planszy (obszar w kratkę, figure) można rozmieszczać (przeciągając) ikonki kontrolek widoczne po lewej stronie.



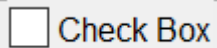
## Kontrolki



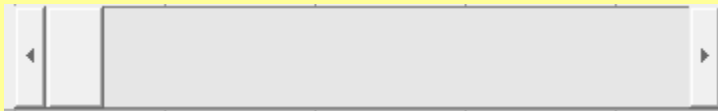
***Push Button*** – przycisk on/off prostokątny



***Radio Button*** - przycisk on/off okrągły, współpraca z *buton panel*



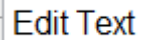
**Check Box** - kontrolka wyboru (zaznaczania)



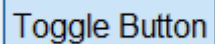
**Slider** – suwak, potencjometr



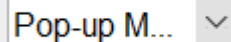
***Static tekst*** - tylko wyświetlanie tekstu



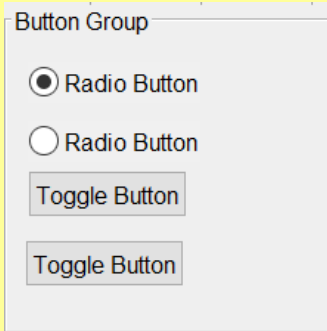
***Edit tekst*** - wpisywanie i wyświetlanie tekstu



***Toggle Button*** - przycisk on/off prostokątny, współpraca z *buton panel*

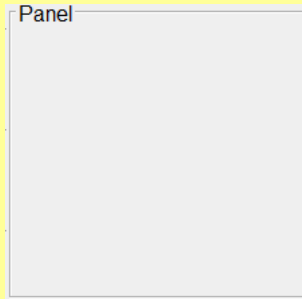


***Pop-up Menu*** – menu rozwijane

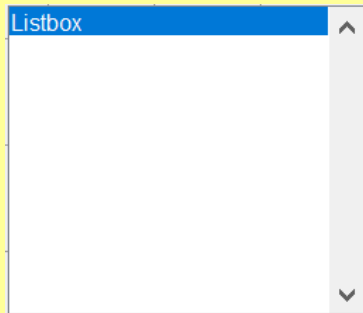


***Button Group*** – zgrupowanie przycisków  
(umożliwia pracę synchroniczną)

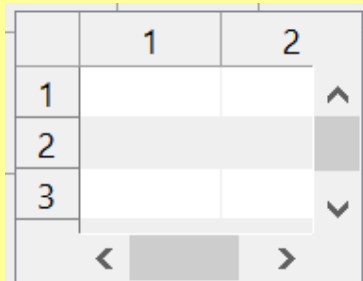
## Kontrolki



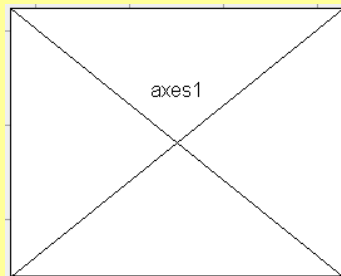
*Panel* – zgrupowanie kontrolki poprawiające przejrzystość



*Listbox* - lista opcji/wartości



*Table* - tabela



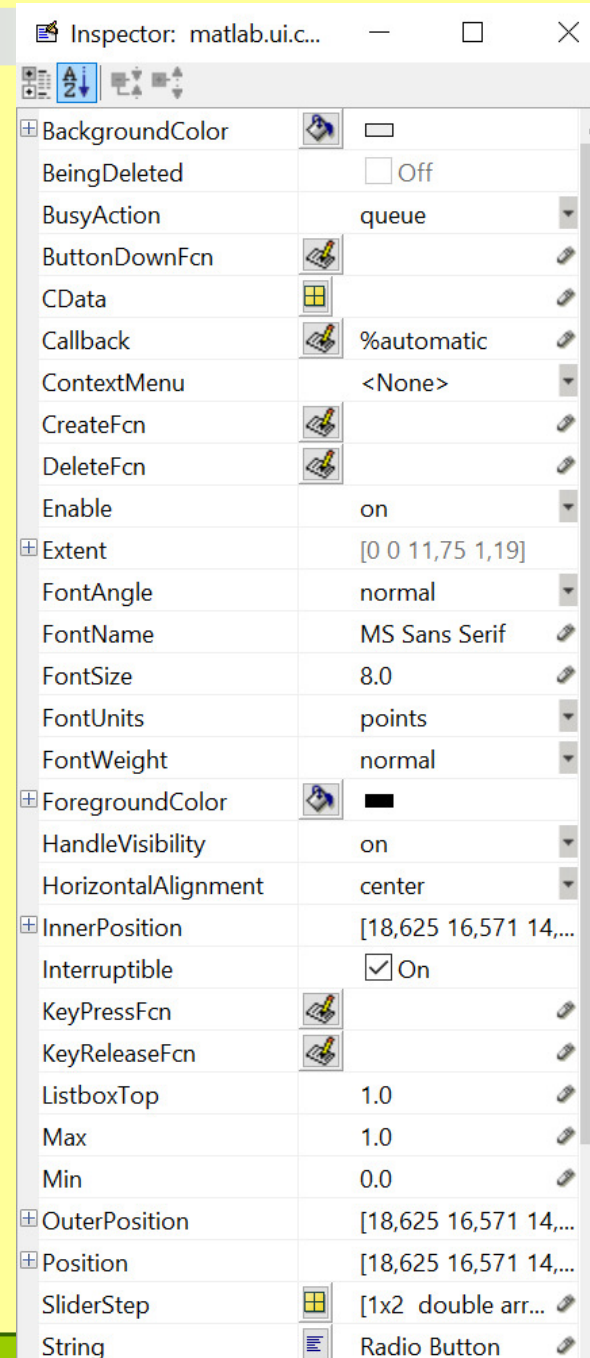
*Axes* – wykres

## Property Inspector

Parametry każdej kontrolki można zobaczyć/zmieniać klikając prawym przyciskiem myszki i wybierając polecenie *Property Inspector*.

Jednym z najważniejszych parametrów kontrolki jest *Tag*, czyli nazwa (etykieta), po której można się do kontrolki odwoływać.

Kolejne ważne parametry to *Value* (wartość numeryczna parametru) i *String* (wypisywany tekst).




## Obsługa zdarzeń

Obsługa zdarzeń w MATLABie odbywa się za pośrednictwem pięciu funkcji:

- CreateFcn
- DeleteFcn
- ButtonDownFcn
- KeyPressFcn
- Callback

Funkcje te są dostępne dla większości kontrolek. W niektórych przypadkach jest dostępna ich tylko część.

Funkcje te są dostępne dla większości kontrolek. W niektórych przypadkach jest dostępna ich tylko część.




1). **CreateFcn** - funkcja ta jest wywoływana w trakcie tworzenia kontrolki ale po ustawieniu wszystkich jej parametrów.

```
function kontrolka1_CreateFcn(hObject, eventdata, handles)
```

- **hObject** - **uchwyt** (identyfikator) do kontrolki kontrolka1
- **eventdata** - zostanie wykorzystane w przyszłych wersjach MATLABa
- **handles** - zbiór **uchwyków** do wszystkich kontrolek - pusty do momentu wywołania wszystkich funkcji CreateFcn

*Uchwyty* to mechanizm wykorzystywany przez MATLABa do identyfikacji kontrolek. Poniższe metody ilustrują sposoby uzyskiwania uchwytów do konkretnych kontrolek:



**handles** - zbiór uchwytów do wszystkich kontrolek. Można go używać wewnątrz funkcji obsługujących zdarzenia. Uzyskanie uchwytu do kontrolki następuje przez wywołanie `handles.tag`, gdzie `tag` oznacza unikalną nazwę kontrolki przypisaną do parametru `Tag`.

`findobj('PropertyName','PropertyValue')` - funkcja zwraca uchwyty kontrolek dla których parametr `PropertyName` posiada wartość `PropertyValue`. W przypadku większej liczby takich kontrolek funkcja zwraca wektor uchwytów.

**hObject** - można go używać wewnątrz funkcji obsługujących zdarzenia. Zawiera uchwyt do kontrolki na rzecz której wywoływana jest dana funkcja.

`gcbo` - uchwyt do kontrolki której `CallBack` jest aktualnie wykonywany.

`gco` - uchwyt do aktywnej kontrolki.

`gca` - uchwyt do aktywnego wykresu.



2). **DeleteFcn** - funkcja ta jest wywoływana w trakcie usuwania kontrolki ale przed usuwaniem jej parametrów.

```
function kontrolka1_DeleteFcn(hObject, eventdata, handles)
```

3). **ButtonDownFcn** - funkcja ta jest wywoływana po kliknięciu myszą na kontrolkę (Gdy parametr Enable == 'on' funkcje wywołuje tylko prawy przycisk myszy. Dla Enable == 'off' funkcję wywołują oba przyciski myszy).

```
function kontrolka1_ButtonDownFcn(hObject, eventdata, handles)
```

4). **KeyPressFcn** - funkcja ta jest wywoływana po naciśnięciu klawisza na aktywnej kontrolce (W przypadku naciśnięcia klawisza spacji wywoływana jest najpierw funkcja KeyPressFcn a następnie Callback. Przy innych klawiszach wywoływana jest tylko funkcja KeyPressFcn).

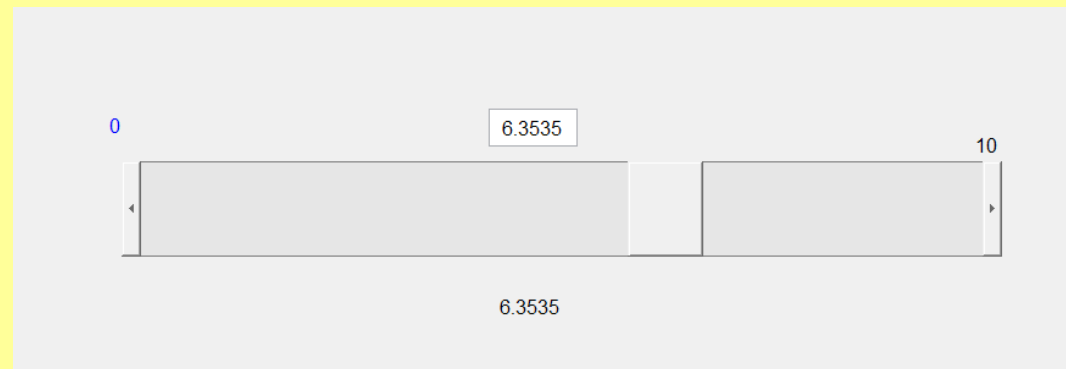
```
function kontrolka1_KeyPressFcn(hObject, eventdata, handles)
```

5). **Callback** - funkcja ta jest wywoływana w sytuacji gdy na danej kontrolce została wykonana właściwa dla niej akcja (dla przycisku - wciśnięcie, radiobutton i checkbox - zaznaczenie, slider - przesunięcie suwaka).

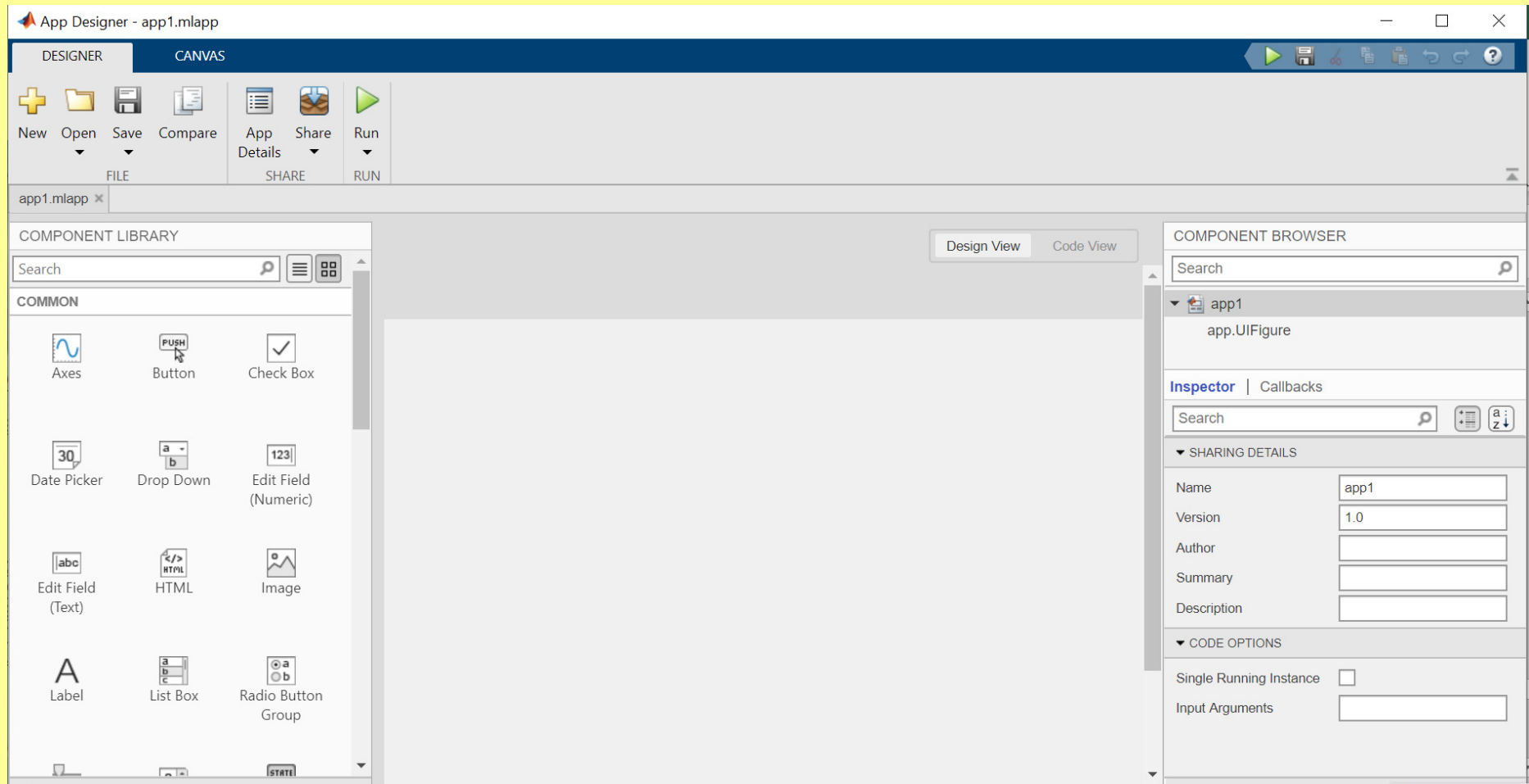
```
function kontrolka1_Callback(hObject, eventdata, handles)
```

## Przykład: Oprogramowanie suwaka

```
function slider_moj_OpeningFcn(hObject, eventdata,  
handles, varargin)  
%wartość minimalna suwaka  
vmin=get(handles.suwak_k, 'Min');  
set(handles.v_min, 'String', num2str(vmin));  
%wartość maksymalna suwaka  
vmax=get(handles.suwak_k, 'Max');  
set(handles.v_max, 'String', num2str(vmax));  
  
function suwak_k_Callback(hObject, eventdata, handles)  
v=get(hObject, 'Value');  
set(handles.tag_stat, 'String', num2str(v));  
set(handles.tag_edit, 'String', num2str(v));
```



# App Designer – następca guide





**DZIĘKUJĘ ZA UWAGĘ**