

GUIプログラムの作成

講師：高安 亮紀

第4回

目的

1. MATLABの「GUIDE」を使い方を学びながら、基本的なGUIプログラムを作成してみましょう。
2. ball1.mの実行結果をGUI上で表示できるように、プログラムを作成してみましょう。

※ GUI (Graphical User Interface) とは、情報表示の際、ボタン・アイコン等のグラフィックスを多用し、マウスで指示できる方式のこと

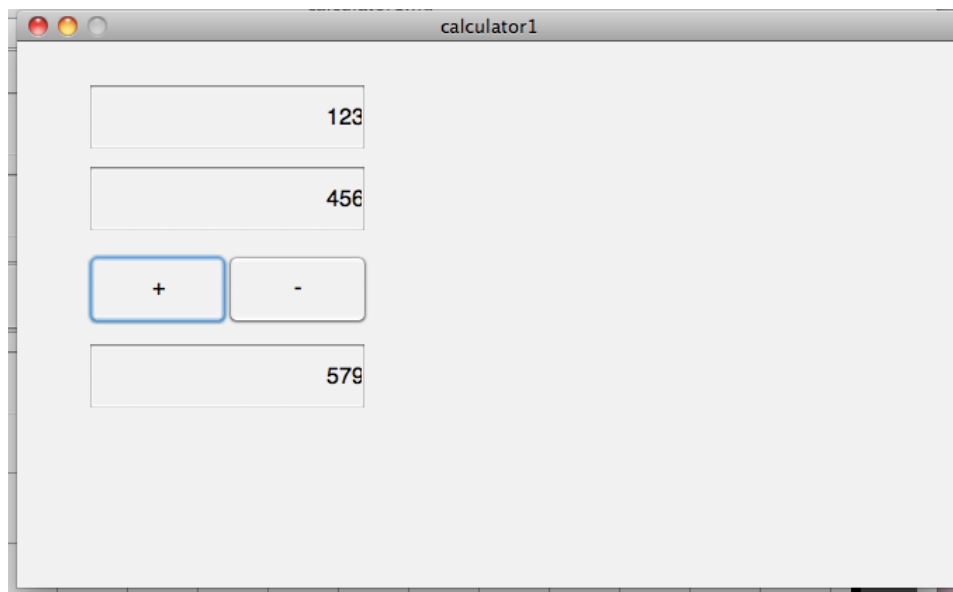
GUIDEとは

- MATLAB上でGUIプログラムを作成するためのシステム
- GUIDE = GUI Development Environment
- Mファイルとfigファイルの組み合わせで1つのプログラムを構成
- Mファイルから、自分で作成した他の関数などを呼び出すこともできる

GUIDEの基本的な使い方

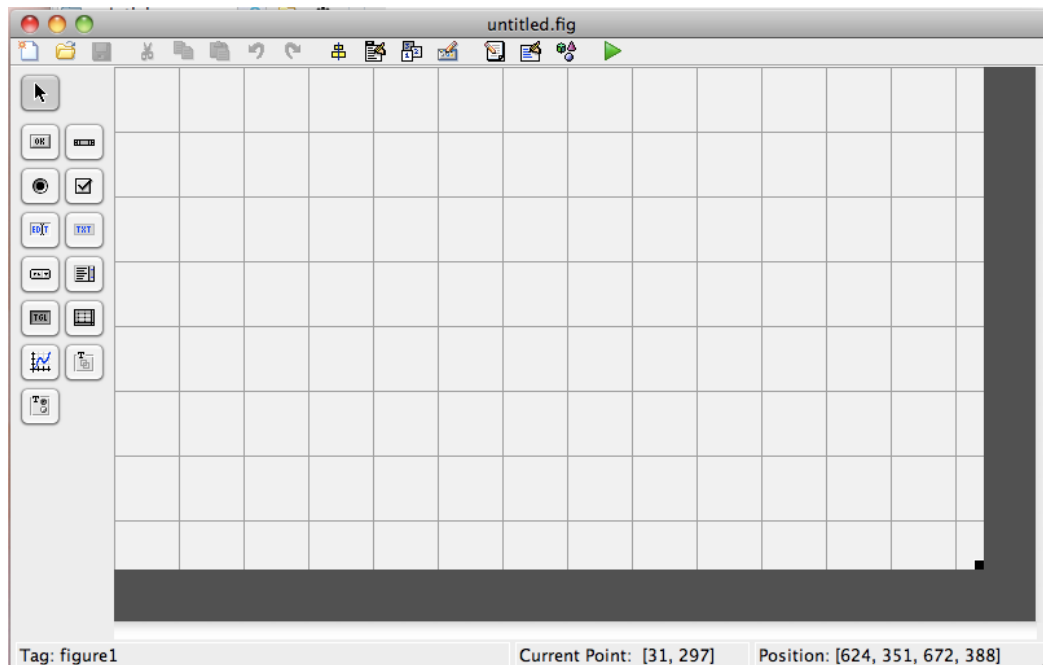
- ・2つの入力枠と2つのボタンを配置し、足し算と引き算の結果が表示されるプログラムを作成しながら、GUIDEの使い方を学びましょう。

<完成イメージ>



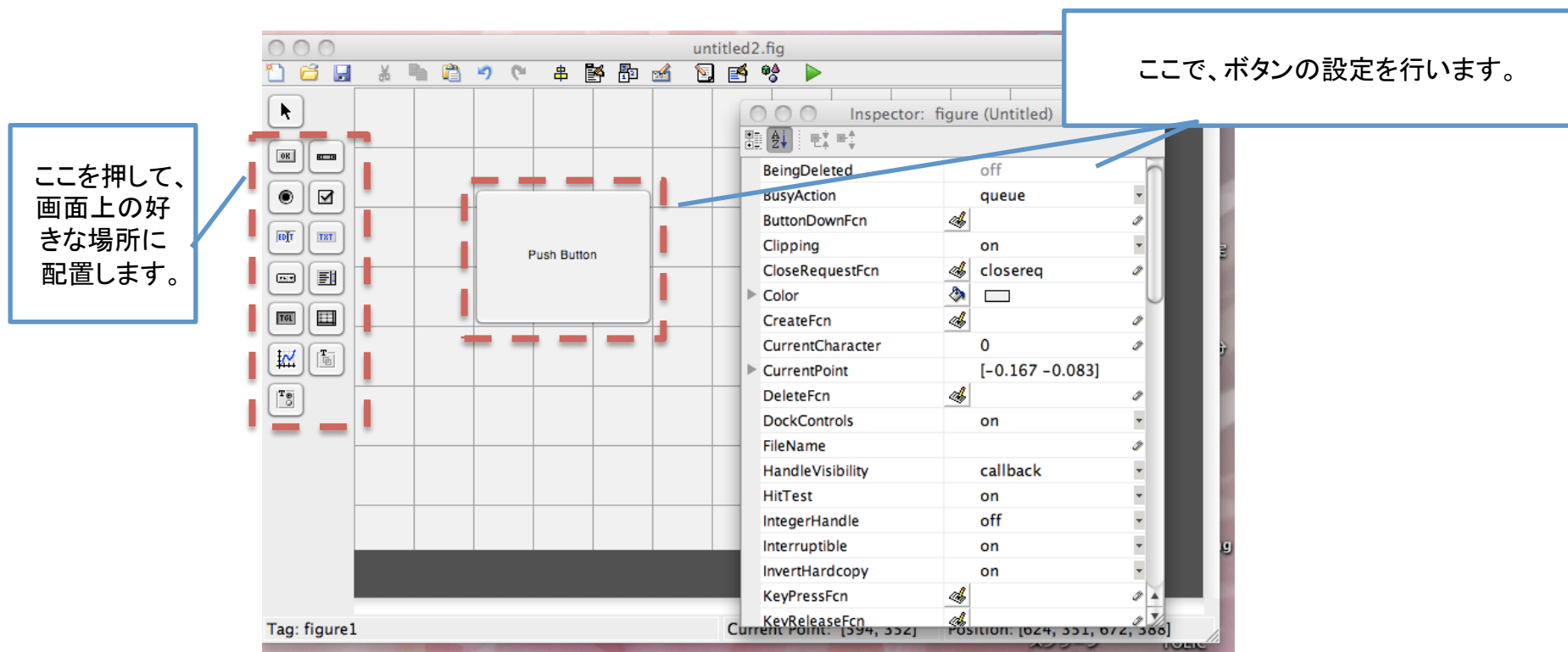
GUIDEの基本的な使い方

- (1) MATLABコマンドウィンドウに `guide` と入力します。
- (2) 「Blank GUI (Default)」を選択すると、
以下が表示されます。



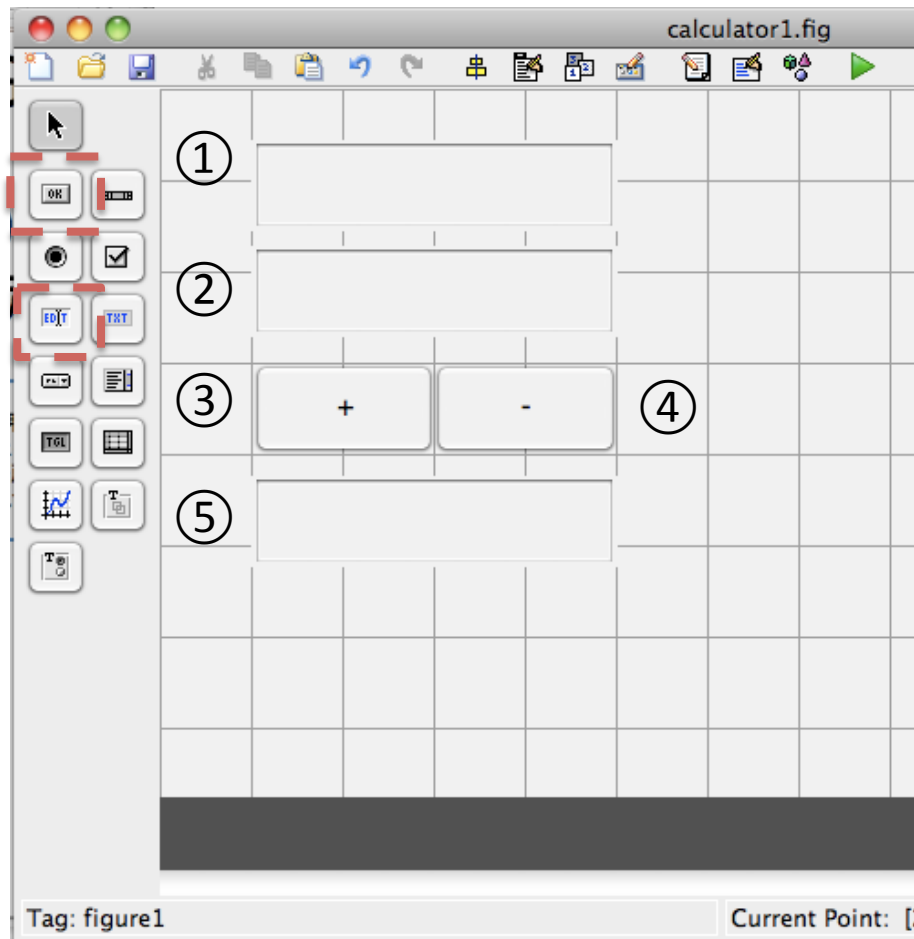
GUIDEの基本的な使い方

(3) 左側のメニューがGUIの部品です。部品を画面上に配置し、クリックしてProperty Inspectorを設定します。



GUIDEの基本的な使い方

(4) テキストボックス (Edit Text) と、ボタンを以下のように配置し、Property Inspectorを設定してください。



①テキストボックス(入力)
String: 空(すべて消す)

Tag: edit1

HorizontalAlignment: right

②テキストボックス(入力)
String: 空(すべて消す)

Tag: edit2

HorizontalAlignment: right

③ボタン(足し算)

String: +

Tag: Pushbutton_plus

④ボタン(引き算)

String: -

Tag: Pushbutton_minus

⑤テキストボックス(出力)
String: 空(すべて消す)

Tag: edit3

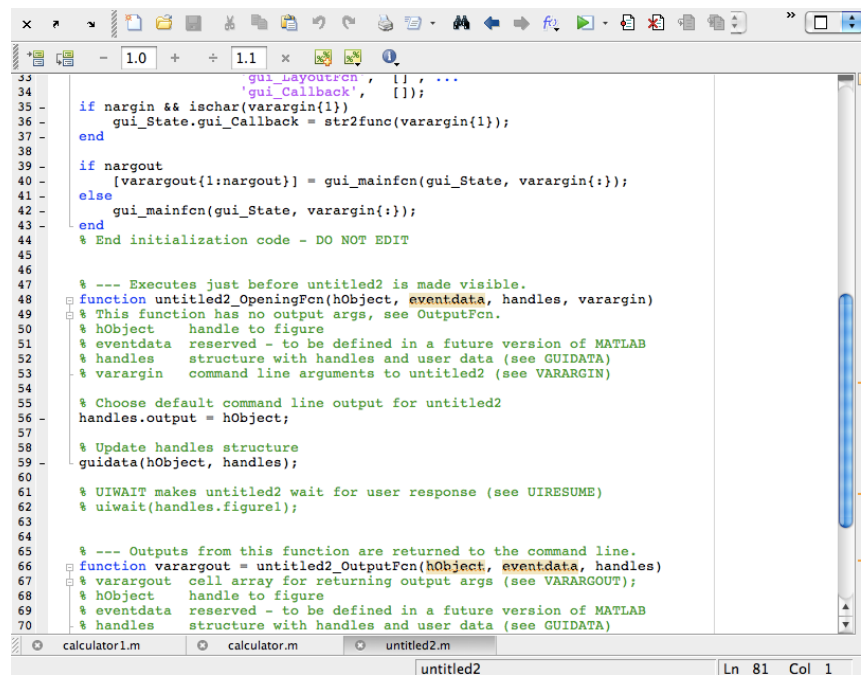
HorizontalAlignment: right

GUIDEの基本的な使い方

(5) File→SAVE ASを押し、「calc1.fig」で保存をします。

(6) 保存をしたタイミングで、「calc1.m」が自動生成される。

calc1.m



```
33     gui_Layoutfcn, [1], ...
34     'gui_Callback', [1]);
35     if nargin && ischar(varargin{1})
36         gui_State.gui_Callback = str2func(varargin{1});
37     end
38
39     if nargout
40         [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
41     else
42         gui_mainfcn(gui_State, varargin{:});
43     end
44     % End initialization code - DO NOT EDIT
45
46
47     % --- Executes just before untitled2 is made visible.
48     function untitled2_OpeningFcn(hObject, eventdata, handles, varargin)
49     % This function has no output args, see OutputFcn.
50     % hObject    handle to figure
51     % eventdata  reserved - to be defined in a future version of MATLAB
52     % handles     structure with handles and user data (see GUIDATA)
53     % varargin    command line arguments to untitled2 (see VARARGIN)
54
55     % Choose default command line output for untitled2
56     handles.output = hObject;
57
58     % Update handles structure
59     guidata(hObject, handles);
60
61     % UIWAIT makes untitled2 wait for user response (see UIRESUME)
62     % uiwait(handles.figure1);
63
64
65     % --- Outputs from this function are returned to the command line.
66     function varargout = untitled2_OutputFcn(hObject, eventdata, handles)
67     % varargout  cell array for returning output args (see VARARGOUT);
68     % hObject    handle to figure
69     % eventdata  reserved - to be defined in a future version of MATLAB
70     % handles     structure with handles and user data (see GUIDATA)
```

GUIDEの基本的な使い方

(7) calc1.mの、テキストボックスのコールバック(テキストを入力したときやボタンが押されたときなどのアクション)を編集します。

edit1, edit2のコールバック関数内に、以下赤字部分を追加。

```
function edit1_Callback(hObject, eventdata, handles)
```

```
...
```

```
handles.edit1 = str2double(get(hObject,'String'));
```

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

```
function edit2_Callback(hObject, eventdata, handles)
```

```
...
```

```
handles.edit2 = str2double(get(hObject,'String'));
```

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

※ handles.edit1 は、構造体handlesの中にあるedit1という変数を意味する。

Property inspectorで編集したTagの名前に合わせる。

※ get(hObject,'String') は、テキストボックスから文字列を取り出す。

※ str2double(str) は、文字列strを数値に変換する。

GUIDEの基本的な使い方

(8) calc1.mの、プッシュボタンのコールバックを編集します。Pushbutton_plus、Pushbutton_minusのコールバック関数内に、以下赤文字部分を追加。

```
function Pushbutton_plus_Callback(hObject, eventdata, handles)
...
```

```
res = handles.edit1 + handles.edit2;
set(handles.edit3,'String',num2str(res));
```

※ num2str(res) は、res(数値)を文字に変換する。

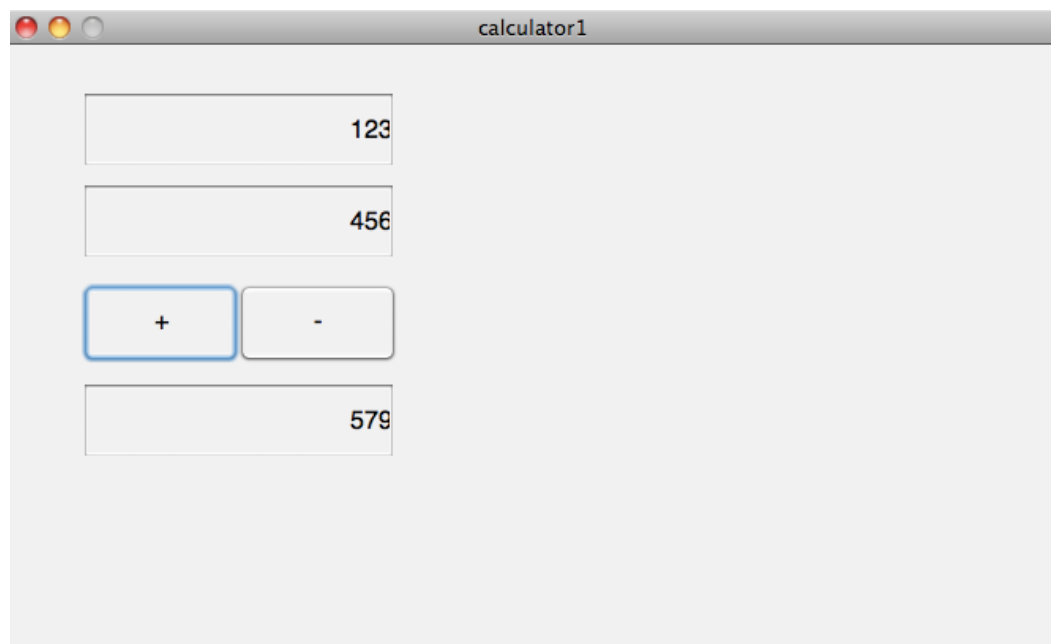
※ set(handles.edit3,'String',str) は、文字strをedit3にセットする。

Pushbutton_minusも同様に追加しましょう。

GUIDEの基本的な使い方

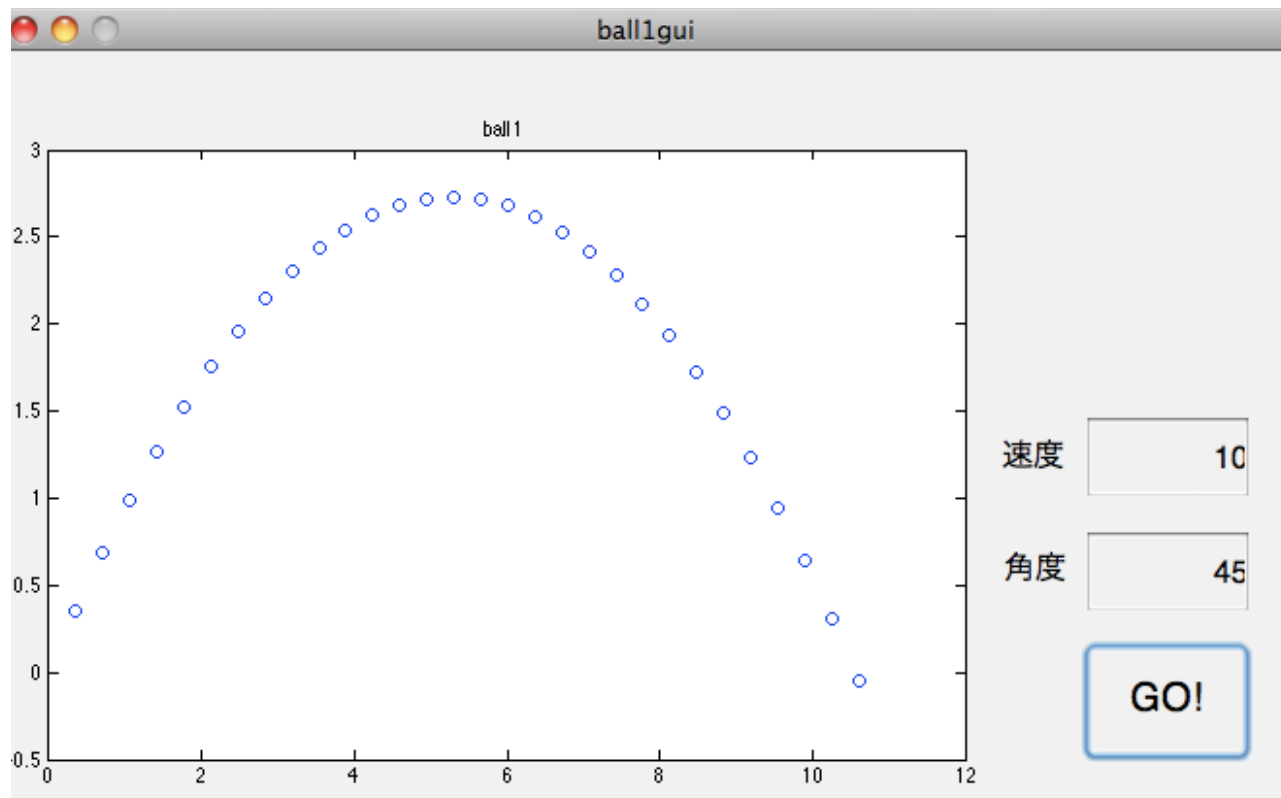
(9) 編集したファイルを保存し、figファイルのウィンドウにある「Run Figure」(再生ボタン)を押すか、コマンドウィンドウ上で、calc1と実行。

実行結果: GUIが表示され、計算が出来ます



演習： ball1.mのGUIプログラムを作成

<完成イメージ>



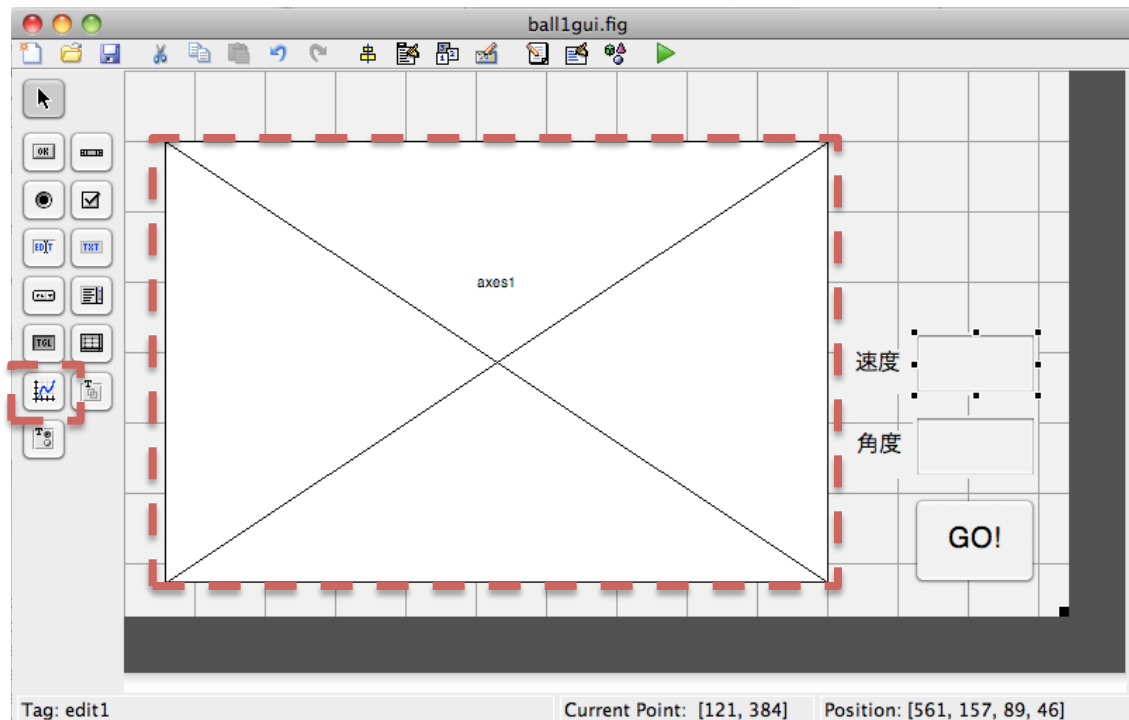
演習：ball1.mのGUIプログラムを作成

- 作成手順

- 1) **部品の配置**：GUIDEを使って、完成イメージのようになるようにテキストボックス、プッシュボタン、座標軸の部品を配置し、ball1_gui.figとして保存する。
- 2) **エディタで編集1**：ball1_gui.mの中で、テキストボックスのコールバック部分を探し、入力された数字をそれぞれ初速度と角度として受け取るようにする。
- 3) **エディタで編集2**：ball1_gui.mの中で、プッシュボタンのコールバック部分を探し、ball1.mのグラフを出力させるように設定（次ページにヒントあり）。
- 4) **エディタでファイルを保存**
- 5) **実行**：コマンドウィンドウで ball1_gui と入力し、GUI上で実行する。

演習ヒント：グラフの表示

- 左メニューのグラフを赤枠を画面上の配置します。



演習ヒント：グラフの表示

- ball1_gui.mのプッシュボタンのコールバックに以下を入力することで、グラフが出力されます

```
function Pushbutton1_Callback(hObject, eventdata, handles)
```

```
~
```

```
v0 = handles.edit1; % テキストボックス1でTagをedit1にした場合
```

```
ang = handles.edit2; % テキストボックス2でTagをedit2にした場合
```

```
ball1(v0,ang);
```