For making GUI GUIDE is used in Matlab.

Typical function used is

function varargout = ECiitkgpMatlab3DView(varargin)

gui\_Singleton = 0;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @ECiitkgpMatlab3DView\_OpeningFcn, ...

'gui\_OutputFcn', @ECiitkgpMatlab3DView\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

There are two GUIs one for the loading of the Dicom Images and one for showing the Dicom image.

Varargin is used for inputing variable number of inputs.

There can be any number of Dicom files so the number should be a variable.

guidata(object\_handle,data)  
data = guidata(object\_handle)

for handling the Dicom data

**GUIDE Uses guidata**   GUIDE uses guidata to store and maintain the handles structure. In a GUIDE GUI code file, do not overwrite the handles structure or your GUI will no longer work. If you need to store data other than handles for your GUI, you can add new fields to the handles structure and safely place your data there.

**Using guidata in a GUIDE GUI**

If you use GUIDE, you do not need to call guihandles to create a structure, because GUIDE generates a handles structure that contains the GUI's handles. You can add your own data to it, for example from within the OpeningFcn template that GUIDE creates:

% --- Executes just before simple\_gui\_tab is made visible.

function my\_GUIDE\_GUI\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to simple\_gui\_tab (see VARARGIN)

% ...

% add some additional data as a new field called numberOfErrors

handles.numberOfErrors = 0;

% Save the change you made to the structure

guidata(hObject,handles)

Notice that you use the input argument hObject in place of gcbo to refer to the object whose callback is executing.

Suppose you needed to access the numberOfErrors field in a push button callback. Your callback code now looks something like this:

% --- Executes on button press in pushbutton1.

function my\_GUIDE\_GUI\_pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% ...

% No need to call guidata to obtain a structure;

% it is provided by GUIDE via the handles argument

handles.numberOfErrors = handles.numberOfErrors + 1;

% save the changes to the structure

guidata(hObject,handles)

**setappdata -**Specify application-defined data

**Syntax**

setappdata(h,'name',value)

**Description**

setappdata(h,'name',value) sets application-defined data for the object with handle h. The application-defined data, which is created if it does not already exist, is assigned the specified name and value. The value can be any type of data.

# getappdata -Value of application-defined data

## Syntax

value = getappdata(h,'name')  
values = getappdata(h)

## Description

value = getappdata(h,'name') gets the value of the application-defined data with the name specified by name, in the object with handle h. If the application-defined data does not exist, the MATLAB software returns an empty matrix in value.

values = getappdata(h) returns all application-defined data for the object with handle h.

# guihandles -Create structure of handles

## Syntax

handles = guihandles(object\_handle)  
handles = guihandles

## Description

handles = guihandles(object\_handle) returns a structure containing the handles of the objects in a figure, using the value of their [Tag](http://www.mathworks.in/help/techdoc/ref/uicontrol_props.html#Tag) properties as the fieldnames, with the following caveats:

* Objects are excluded if their Tag properties are empty, or are not legal variable names.
* If several objects have the same Tag, that field in the structure contains a vector of handles.
* Objects with hidden handles are included in the structure.

handles = guihandles returns a structure of handles for the current figure.

# gcbo -Handle of object whose callback is executing

## Syntax

h = gcbo  
[h,figure] = gcbo

## Description

h = gcbo returns the handle of the graphics object whose callback is executing.

[h,figure] = gcbo returns the handle of the current callback object and the handle of the figure containing this object.

1. Showmenu function is used for showing the initial menu.
2. Addmenus function is used for adding other menus. Each menu element have their childrens. Thery can be only two stage of childrens.