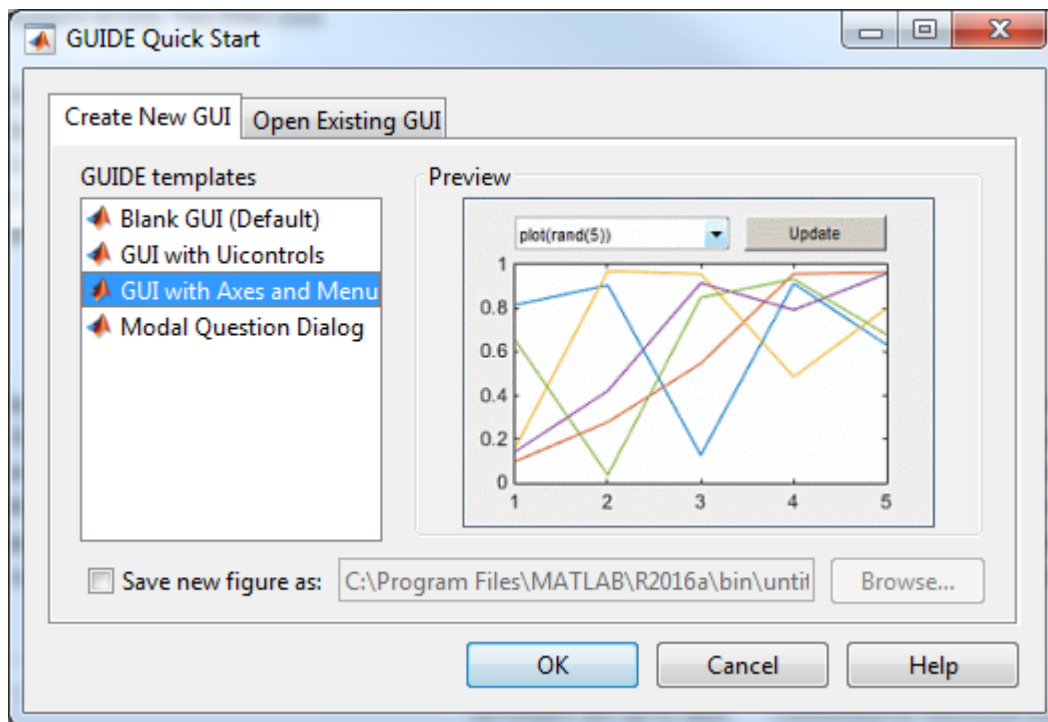


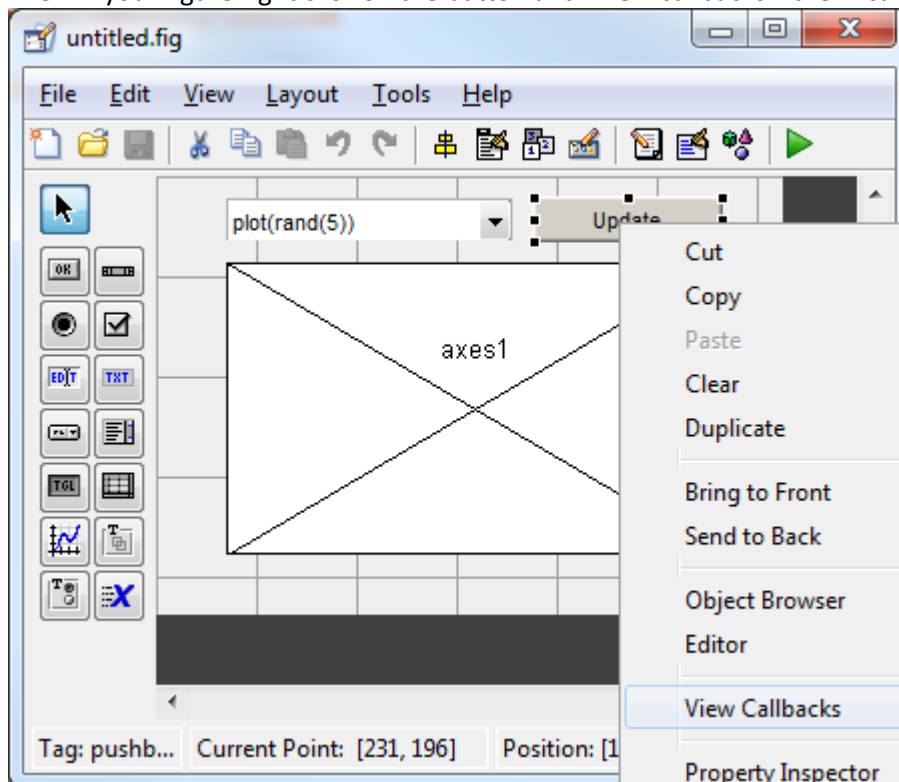
The GUI can be quite simple. I have always made my GUI with GUIDE (not app designer)

Here is one that took my 15 minutes. You would simply type “guide” in matlab, then you could use a default one



Then click ok

Then in your figure right click on the button and “view callbacks “ then “callback”



You can then make things happen on the axis

In the callback you could delete the code that is there and put something like this, which makes a UR5 robot and loads it into the axis

```
axes(handles.axes1);  
L1 = Link('d',0.0892,'a',0,'alpha',-pi/2,'offset',0,'qlim',[deg2rad(-360),deg2rad(360)]);  
L2 = Link('d',0.1357,'a',0.425,'alpha',-pi,'offset',-pi/2,'qlim',[deg2rad(-90),deg2rad(90)]);  
L3 = Link('d',0.1197,'a',0.39243,'alpha',pi,'offset',0,'qlim',[deg2rad(-170),deg2rad(170)]);
```

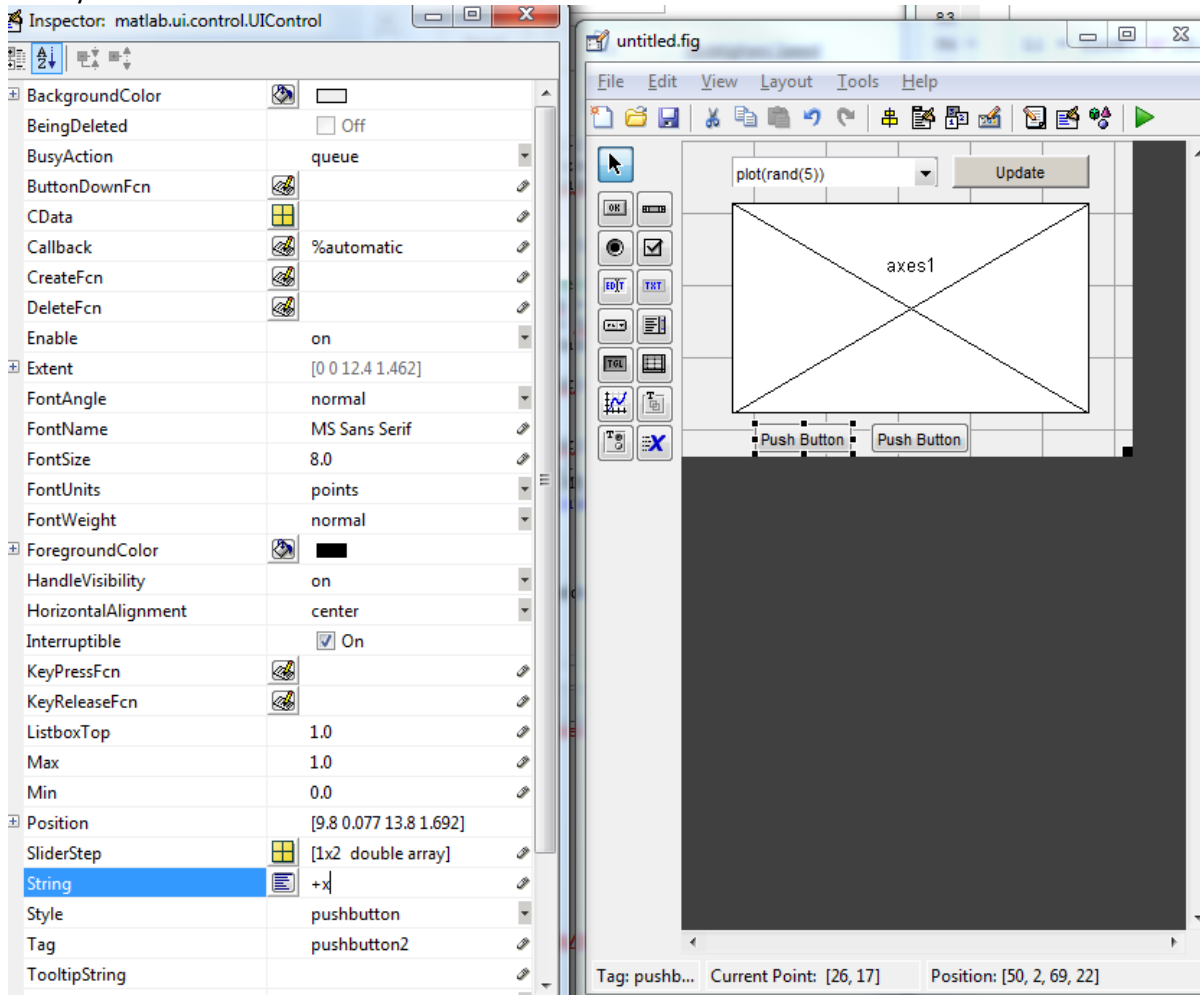
```

L4 = Link('d',0.093,'a',0,'alpha',-pi/2,'offset',-pi/2,'qlim',[deg2rad(-360),deg2rad(360)]);
L5 = Link('d',0.093,'a',0,'alpha',-pi/2,'offset',0,'qlim',[deg2rad(-360),deg2rad(360)]);
L6 = Link('d',0,'a',0,'alpha',0,'offset',0,'qlim',[deg2rad(-360),deg2rad(360)]);
model = SerialLink([L1 L2 L3 L4 L5 L6],'name','UR5');
for linkIndex = 0:model.n
    [ faceData, vertexData, plyData{linkIndex+1} ] =
plyread(['UR5Link',num2str(linkIndex),'.ply'],'tri'); %#ok<AGROW>
    model.faces{linkIndex+1} = faceData;
    model.points{linkIndex+1} = vertexData;
end
% Display robot
workspace = [-2 2 -2 2 -0.3 2];
model.plot3d(zeros(1,model.n),'noarrow','workspace',workspace);
if isempty(findobj(get(gca,'Children'),'Type','Light'))
    camlight
end
model.delay = 0;
% Try to correctly colour the arm (if colours are in ply file data)
for linkIndex = 0:model.n
    handles = findobj('Tag', model.name);
    h = get(handles,'UserData');
    try
        h.link(linkIndex+1).Children.FaceVertexCData = [plyData{linkIndex+1}.vertex.red ...
                                                            , plyData{linkIndex+1}.vertex.green ...
                                                            , plyData{linkIndex+1}.vertex.blue]/255;

        h.link(linkIndex+1).Children.FaceColor = 'interp';
    catch ME_1
        disp(ME_1);
        continue;
    end
end
data = guidata(hObject);
data.model = model;
guidata(hObject,data);

```

Then you could add some more buttons to move the robot. Here I have added a +x and -x move button



Then in the +x and -x move functions you could have something like this to move in the +x and then just change the +0.01 for -0.01 for the move in the -x

```
function plusX_pushbutton_Callback(hObject, eventdata, handles)
q = handles.model.getpos;
tr = handles.model.fkine(q);
tr(1,4) = tr(1,4) + 0.01;
newQ = handles.model.ikcon(tr,q);
handles.model.animate(newQ);
```

BTW, to do the advanced teach you only need to do this for +y and +z and the movement of the joints 1-6, then it is the gui basically done.

You should also add an estop and then you can add other functions to do interesting things. Please find attached the code and the gui mat file.

