

Building GUIs in MATLAB

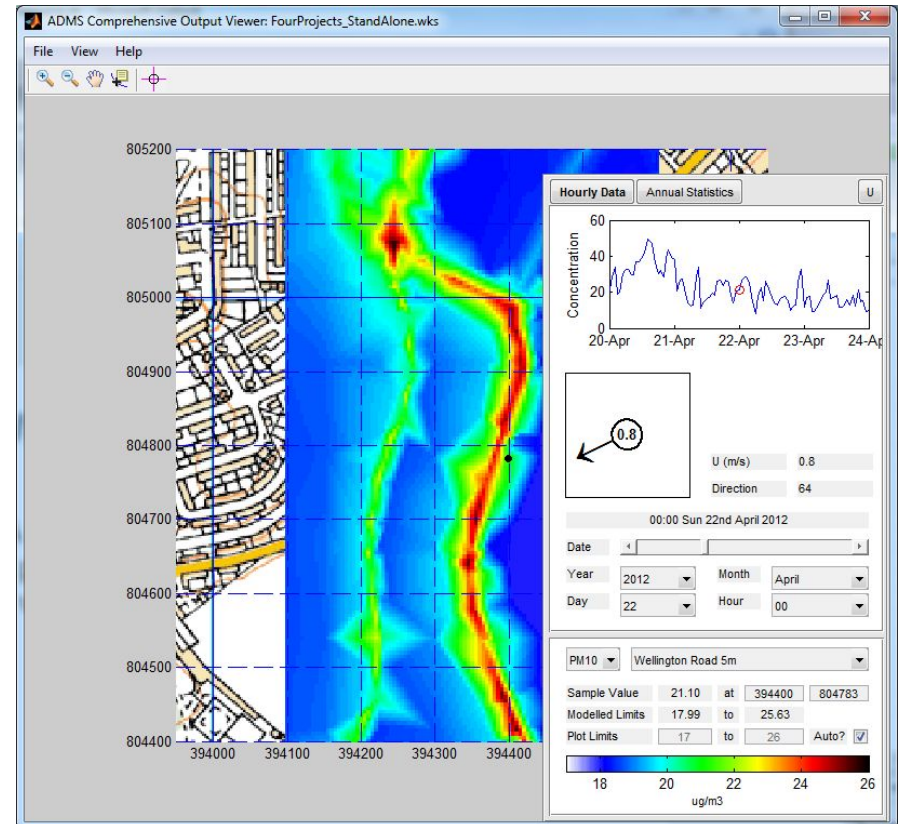
Eddy Barratt
16th December 2016

- **Why use GUIs?**
- **Object Oriented Programming**
- **An example GUI Class**
 - **Constructor**
 - **Callbacks**
 - **In Actions.**

Why GUIs?

A well designed Graphical User Interface (GUI) provides an intuitive way for a user to interact with a complex computer programme.

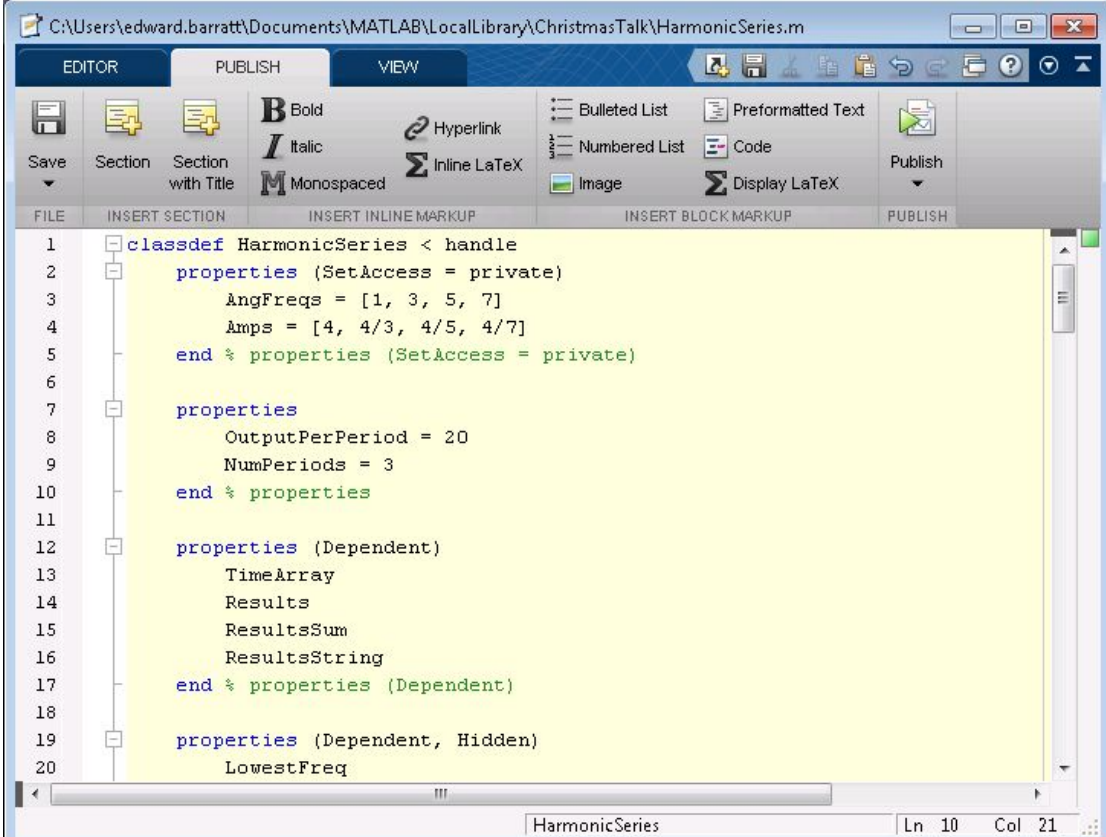
Stand by for a MATLAB GUI demonstration...



Object Oriented Programming

Object Oriented programming involves creating classes that contain properties and methods related to the objects they represent.

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action



The screenshot shows a MATLAB editor window with the file path `C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeries.m`. The editor displays a class definition for `HarmonicSeries` using the `classdef` keyword. The class is defined as a handle class with three property sets: a private set for angular frequencies and amplitudes, a standard set for output and period parameters, and a dependent/hidden set for time array, results, and lowest frequency.

```
1 classdef HarmonicSeries < handle
2     properties (SetAccess = private)
3         AngFreqs = [1, 3, 5, 7]
4         Amps = [4, 4/3, 4/5, 4/7]
5     end % properties (SetAccess = private)
6
7     properties
8         OutputPerPeriod = 20
9         NumPeriods = 3
10    end % properties
11
12    properties (Dependent)
13        TimeArray
14        Results
15        ResultsSum
16        ResultsString
17    end % properties (Dependent)
18
19    properties (Dependent, Hidden)
20        LowestFreq
```

Object Oriented Programming

Watch as I code...

```
>> HS = HarmonicSeries
```

```
HS =
```

```
HarmonicSeries with properties:
```

```
    AngFreqs: [1 3 5 7]
      Amps: [4 1.3333 0.8000]
0.5714]
    OutputPerPeriod: 20
      NumPeriods: 3
      TimeArray: [1x421 double]
      Results: [4x421 double]
      ResultsSum: [4x421 double]
      ResultsString: [1x59 char]
```

```
>> HS = HarmonicSeries([1,2,3])
```

```
HS =
```

```
HarmonicSeries with properties:
```

```
    AngFreqs: [1 2 3]
      Amps: [1 1 1]
    OutputPerPeriod: 20
      NumPeriods: 3
      TimeArray: [1x181 double]
      Results: [3x181 double]
      ResultsSum: [3x181 double]
      ResultsString: 'y = 1sin(1x) +
1sin(2x) + 1sin(3x)'
```

```
>> HS = HarmonicSeries([1,2,3], [4, 2, 4])
```

```
HS =
```

```
HarmonicSeries with properties:
```

```
    AngFreqs: [1 2 3]
      Amps: [4 2 4]
    OutputPerPeriod: 20
      NumPeriods: 3
      TimeArray: [1x181 double]
      Results: [3x181 double]
      ResultsSum: [3x181 double]
      ResultsString: 'y = 4sin(1x) + 4sin(3x) +
2sin(2x)'
```

```
>> HS.AngFreqs = [23, 45]
```

You cannot set the read-only property 'AngFreqs' of HarmonicSeries.

```
>> HS.SetFreqsAndAmps([23, 45], [5, 6])
```

```
>> HS
```

```
HS =
```

```
HarmonicSeries with properties:
```

```
    AngFreqs: [23 45]
      Amps: [5 6]
    OutputPerPeriod: 20
      NumPeriods: 3
      TimeArray: [1x118 double]
      Results: [2x118 double]
      ResultsSum: [2x118 double]
      ResultsString: 'y = 6sin(45x) + 5sin(23x)'
```

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action

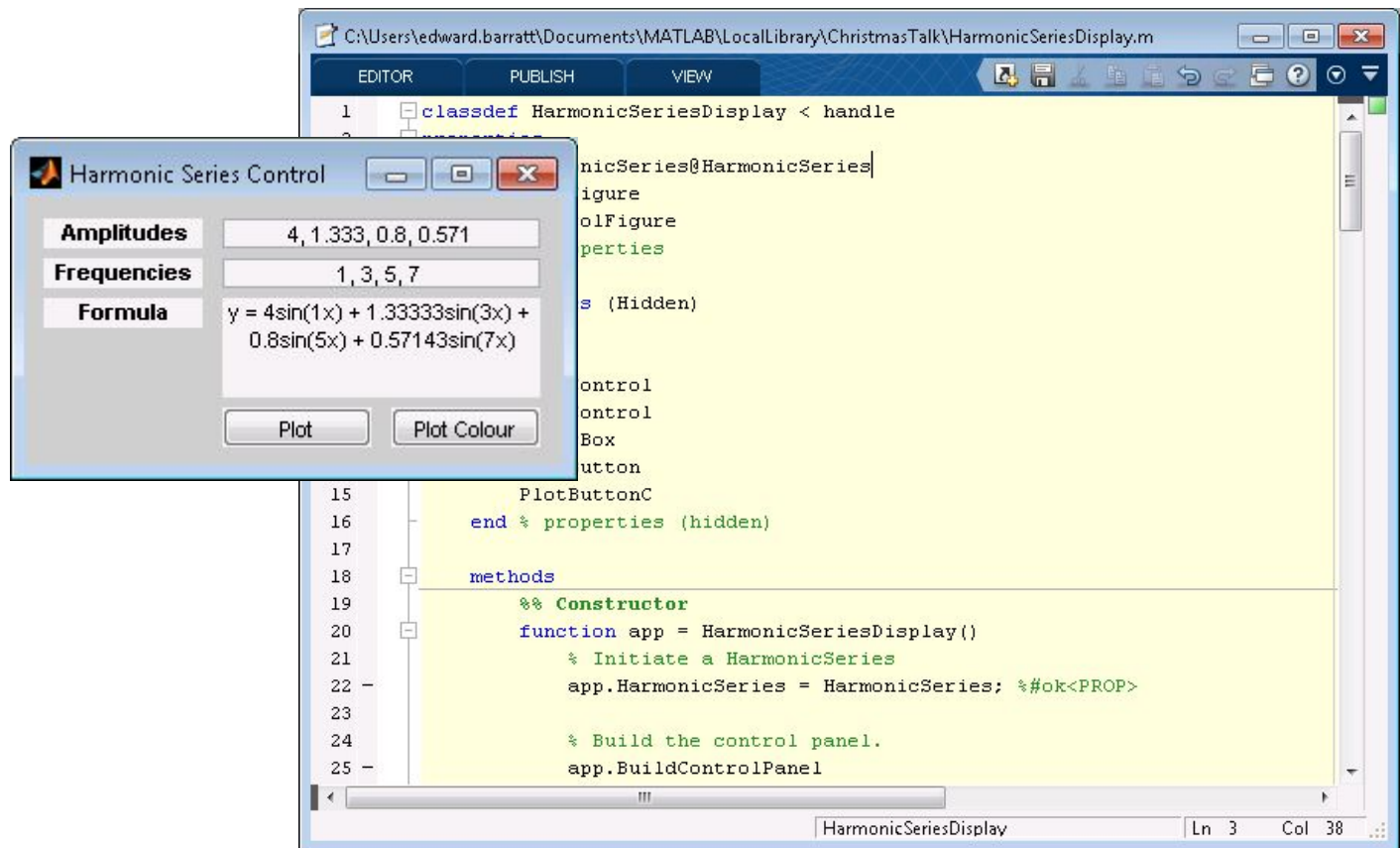
Object Oriented Programming

- Classes are self contained objects with properties and methods.
- Classes can be called by other programmes, or stand alone.

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action

A GUI Class

- Classes can raise, control, and be controlled by figures.



A GUI Class

Constructor

Constructor defines the
“HarmonicSeries” property
by initialising a
HarmonicSeries object.

Then it calls the
“BuildControlPanel”
method

Followed by the
“FillControlPanel”
method

```

14     PlotButton
15     PlotButtonC
16 end % properties (hidden)
17
18 methods
19     %% Constructor
20     function app = HarmonicSeriesDisplay()
21         % Initiate a HarmonicSeries
22         app.HarmonicSeries = HarmonicSeries; %#<<PROP>
23
24         % Build the control panel
25         app.BuildControlPanel
26         app.FillControlPanel
27     end % function app = HarmonicSeriesDisplay()
28
29     function PlotSeries(app, Sender, ~)
30         % Read control, and set HarmonicSeries
31         app.ReadAndSet
32
33         % Get the figure, or create it.
34         if ~isempty(app.PlotFigure) && ishghandle(app.PlotFigure)
35             figure(app.PlotFigure);
36         else
37             app.PlotFigure = figure();
38         end
39     end

```


A GUI Class - uicontrol

BuildControlPanel method

Define the ControlFigure property, which is a MATLAB figure object.

```

66 function BuildControlPanel(app)
67
68     app.ControlFigure = figure('Visible', 'off', ...
69         'Position', [100, 100, 270, 135], ...
70         'MenuBar', 'none', ...
71         'Name', 'Harmonic Series Control', ...
72         'NumberTitle', 'off');
73     movegui(app.ControlFigure, 'center')
74
75     % Amplitudes
76     uicontrol('style', 'text', ...
77         'Position', [10, 110, 80, 15], ...
78         'String', 'Amplitudes', ...
79         'FontWeight', 'bold')
80     app.AmpsControl = uicontrol('style', 'edit', ...
81         'Position', [100, 110, 160, 15], ...
82         'String', '-----');
83
84     % Frequencies
85     uicontrol('style', 'text', ...
86         'Position', [10, 90, 80, 15], ...
87         'String', 'Frequencies', ...
88         'FontWeight', 'bold')
89     app.FreqControl = uicontrol('style', 'edit', ...
90         'Position', [100, 90, 160, 15], ...
  
```

Define various User Interface Controls and place them within the figure.

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action

A GUI Class - uicontrol

BuildControlPanel method

Define the ControlFigure property, which is a MATLAB figure object.

The image shows a MATLAB script editor window with the following code:

```

66 function BuildControlPanel(app)
67
68     app.ControlFigure = figure('Vis
69         'Position', [100, 100, 270,
70         'MenuBar', 'none', ...
71         'Name', 'Harmonic Series Co
72         'NumberTitle', 'off');
73     movegui(app.ControlFigure, 'cen
74
75     % Amplitudes
76     uicontrol('style', 'text', ...
77         'Position', [10, 110, 80, 15], ...
78         'String', 'Amplitudes', ...
79         'FontWeight', 'bold')
80     app.AmpsControl = uicontrol('style', 'edit', ...
81         'Position', [100, 110, 160, 15], ...
82         'String', '-----');
83
84     % Frequencies
85     uicontrol('style', 'text', ...
86         'Position', [10, 90, 80, 15], ...
87         'String', 'Frequencies', ...
88         'FontWeight', 'bold')
89     app.FreqControl = uicontrol('style', 'edit', ...
90         'Position', [100, 90, 160, 15], ...

```

Overlaid on the code is a window titled "Harmonic Series Control". The window contains three input fields labeled "Amplitudes", "Frequencies", and "Formula", each followed by a text box containing "-----". At the bottom of the window are two buttons labeled "Plot" and "Plot Colour".

Arrows point from the code to the GUI window: one from line 68 to the window title bar, and others from lines 76-79 and 85-88 to the "Amplitudes" and "Frequencies" labels and text boxes respectively.

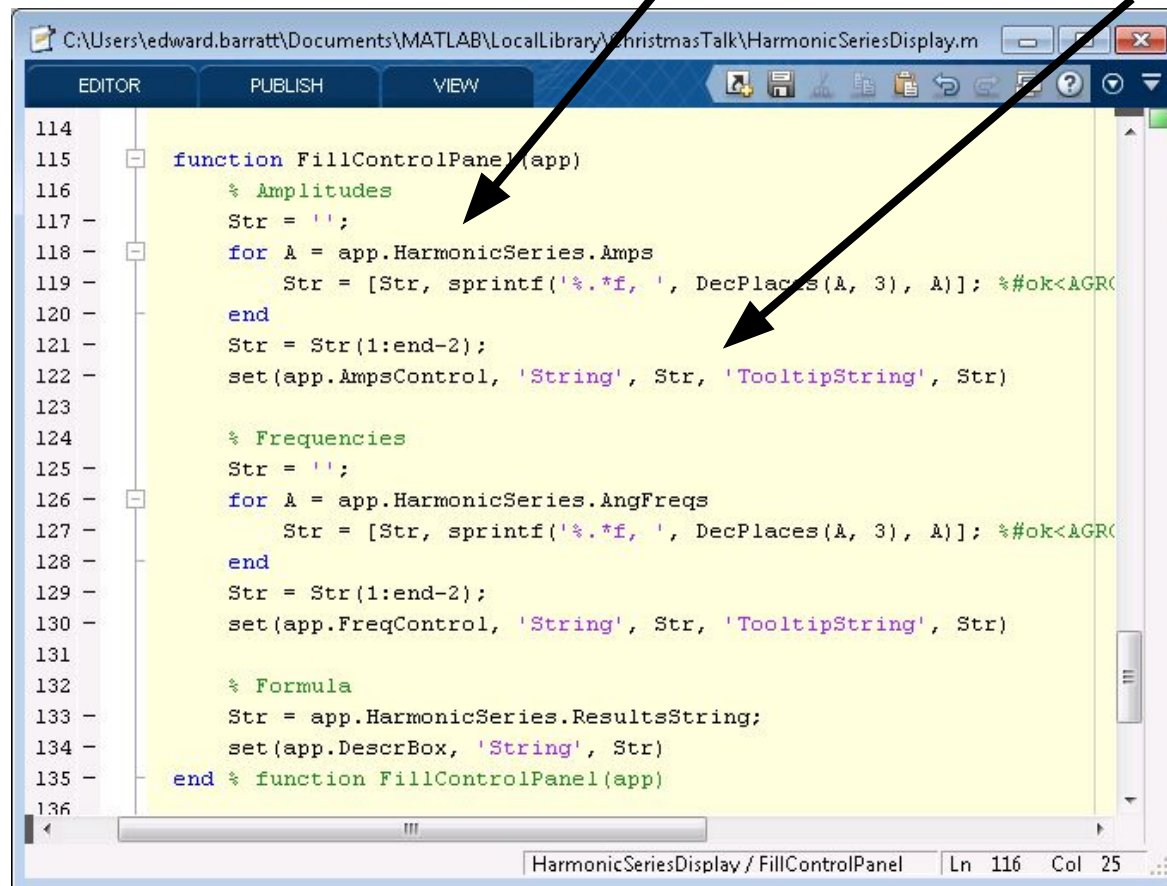
Interface controls and place them within the figure.

A GUI Class - uicontrol

FillControlPanel method

Use the values from the
HarmonicSeries object to
define strings.

And set the 'String'
properties of the
uicontrol objects.



```

114
115 function FillControlPanel(app)
116     % Amplitudes
117     Str = '';
118     for A = app.HarmonicSeries.Amps
119         Str = [Str, sprintf('%.2f, ', DecPlaces(A, 3), A)]; %#ok<AGR
120     end
121     Str = Str(1:end-2);
122     set(app.AmpsControl, 'String', Str, 'TooltipString', Str)
123
124     % Frequencies
125     Str = '';
126     for A = app.HarmonicSeries.AngFreqs
127         Str = [Str, sprintf('%.2f, ', DecPlaces(A, 3), A)]; %#ok<AGR
128     end
129     Str = Str(1:end-2);
130     set(app.FreqControl, 'String', Str, 'TooltipString', Str)
131
132     % Formula
133     Str = app.HarmonicSeries.ResultsString;
134     set(app.DescrBox, 'String', Str)
135 end % function FillControlPanel(app)
136

```

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action

A GUI Class - uicontrol

FillControlPanel method

Use the values from the
HarmonicSeries object to
define strings.

And set the 'String'
properties of the
uicontrol objects.

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action

The image shows a MATLAB Editor window with the `FillControlPanel` function and a running GUI window titled "Harmonic Series Control".

Code in MATLAB Editor:

```

114
115 function FillControlPanel(app)
116     % Amplitudes
117     Str = '';
118     for A = app.HarmonicSeries.Amps
119         Str = [Str, sprintf('%.2f, ', D
120     end
121     Str = Str(1:end-2);
122     set(app.AmpsControl, 'String', Str,
123
124     % Frequencies
125     Str = '';
126     for A = app.HarmonicSeries.AngFreqs
127         Str = [Str, sprintf('%.2f, ', D
128     end
129     Str = Str(1:end-2);
130     set(app.FreqControl, 'String', Str,
131
132     % Formula
133     Str = app.HarmonicSeries.ResultsString;
134     set(app.DescrBox, 'String', Str)
135 end % function FillControlPanel(app)
136

```

Harmonic Series Control GUI Window:

Amplitudes	4, 1.333, 0.8, 0.571
Frequencies	1, 3, 5, 7
Formula	$y = 4\sin(1x) + 1.33333\sin(3x) + 0.8\sin(5x) + 0.57143\sin(7x)$
<input type="button" value="Plot"/> <input type="button" value="Plot Colour"/>	

Arrows indicate that the `Str` variable in the code is populated with data from the `HarmonicSeries` object and then used to set the `'String'` property of the GUI controls.

A GUI Class - callbacks

BuildControlPanel method

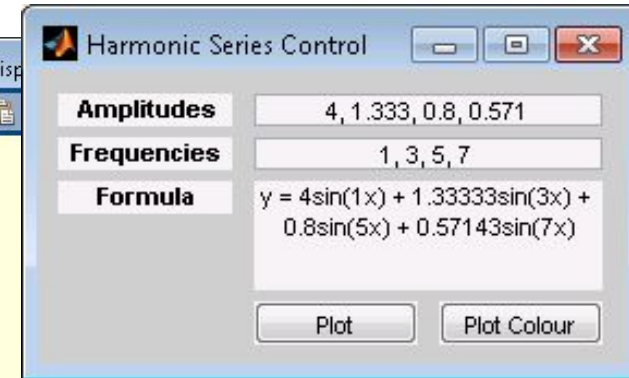
Two different 'pushbutton' uicontrols to plot the harmonic series.

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action

```

92
93 % Description
94 - uicontrol('style','text', ...
95     'Position',[10, 70, 80, 15], ...
96     'String','Formula', ...
97     'FontWeight','bold')
98 - app.DescrBox = uicontrol('style','text', ...
99     'Position',[100, 35, 160, 50], ...
100     'String','-----');
101
102 % Plot Buttons
103 - app.PlotButton = uicontrol('style','pushbutton', ...
104     'Position',[100, 10, 75, 20], ...
105     'String','Plot', ...
106     'Callback', @app.PlotSeries);
107 - app.PlotButtonC = uicontrol('style','pushbutton', ...
108     'Position',[185, 10, 75, 20], ...
109     'String','Plot Colour', ...
110     'Callback', @app.PlotSeries);
111
112 - set(app.ControlFigure, 'Visible', 'on')
113 - end % function BuildControlPanel(app)
114

```



Both calling the same 'PlotSeries' method.

A GUI Class - callbacks

PlotSeries method

```

28
29
30 function PlotSeries(app, Sender, ~)
31     % Read control, and set HarmonicSeries
32     try
33         app.ReadAndSet
34     catch err
35         msgbox(err.message)
36         app.FillControlPanel
37         return
38     end
39
40     % Get the figure, or create it.
41     if isempty(app.PlotFigure) && ishghandle(app.PlotFigure)
42         figure(app.PlotFigure);
43     else
44         app.PlotFigure = figure();
45     end
46
47     app.Plt = plot(app.HarmonicSeries.TimeArrayPi, app.HarmonicSeries.ResultsSum);
48     NumLines = numel(app.Plt);
49     Shades = linspace(0.5, 0.1, NumLines-1);
50     Colors = {'m', 'r', 'g', 'c', 'b'};
51     ci = 0;
52     for linei = 1:NumLines
53         line = app.Plt(linei);
54         ci = ci + 1;
55         if ci > NumLines
56             ci = 1;
57         end
58         if linei ~= NumLines
59             switch Sender
60                 case app.PlotButton
61                     set(line, 'Color', Shades(linei)*[1,1,1], 'linewidth', 1)
62                     otherwise
63                         set(line, 'Color', Colors{ci}, 'linewidth', 1)
64             end
65         else
66             set(line, 'Color', 'k', 'linewidth', 2)
67         end
68     end
69     xlabel('x')
70     ylabel('y')
71 end % function PlotSeries(app)

```

Read the contents of the GUI and set the HarmonicSeries object to take the appropriate values.

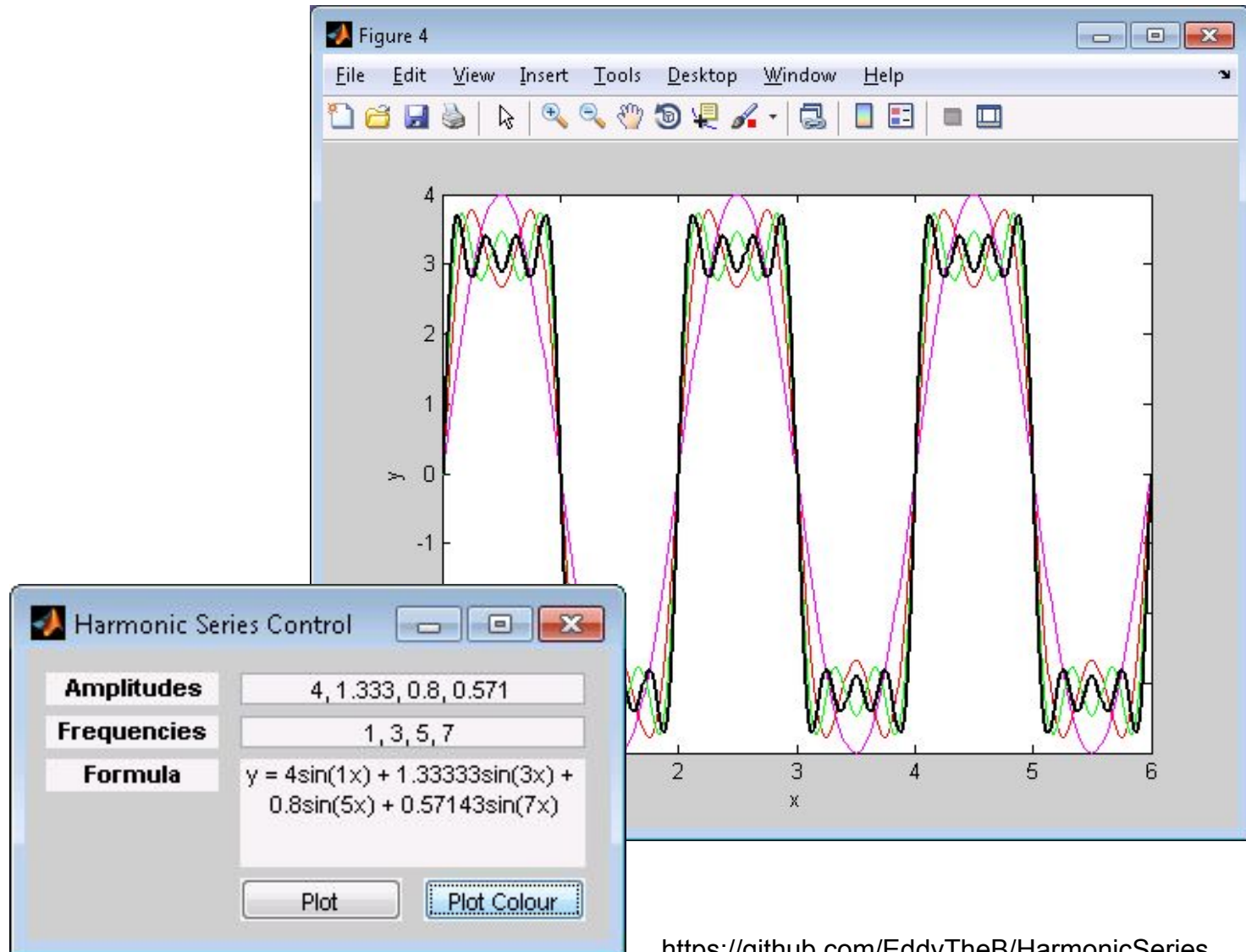
Open a figure, or get it if it's already been opened.

Plot the lines.

The "Sender" argument is used to control the line colours

A GUI Class - In Action

- Why GUIs?
- Object Oriented
- A GUI Class
- Constructor
- Callbacks
- In Action



<https://github.com/EddyTheB/HarmonicSeries>