SEPA
Scottish Environment
Protection Agency

# Building GUIs in MATLAB

Eddy Barratt
16th December 2016

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

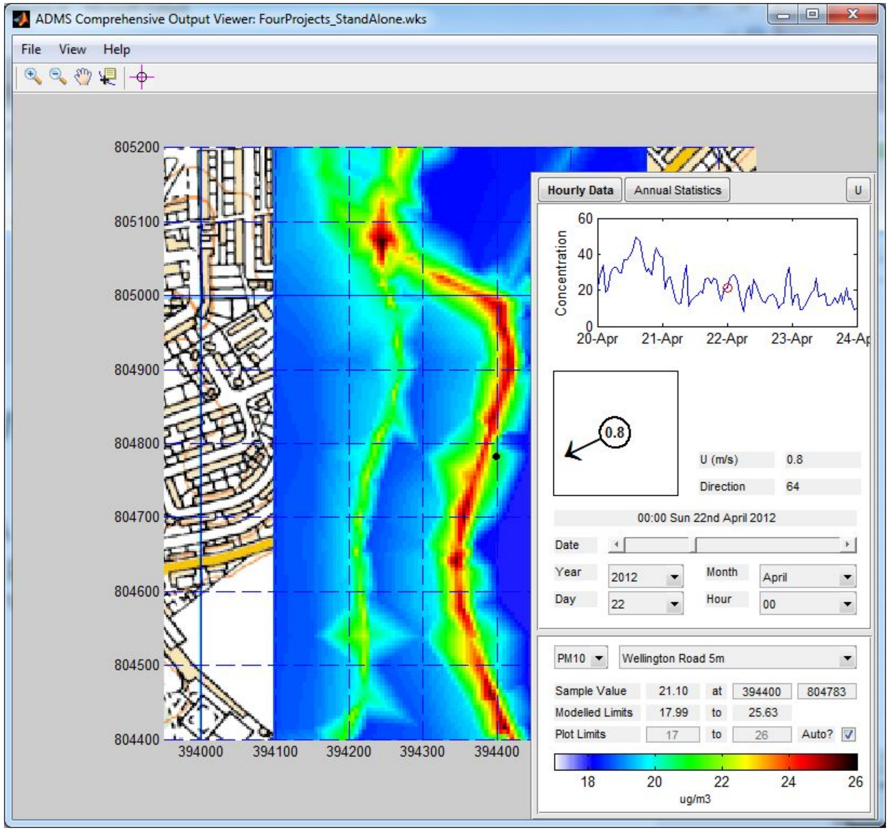
SEPA
Scottish Environment
Protection Agency

# Why GUIs?

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

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...


ADMS Comprehensive Output Viewer: FourProjects_StandAlone.wks

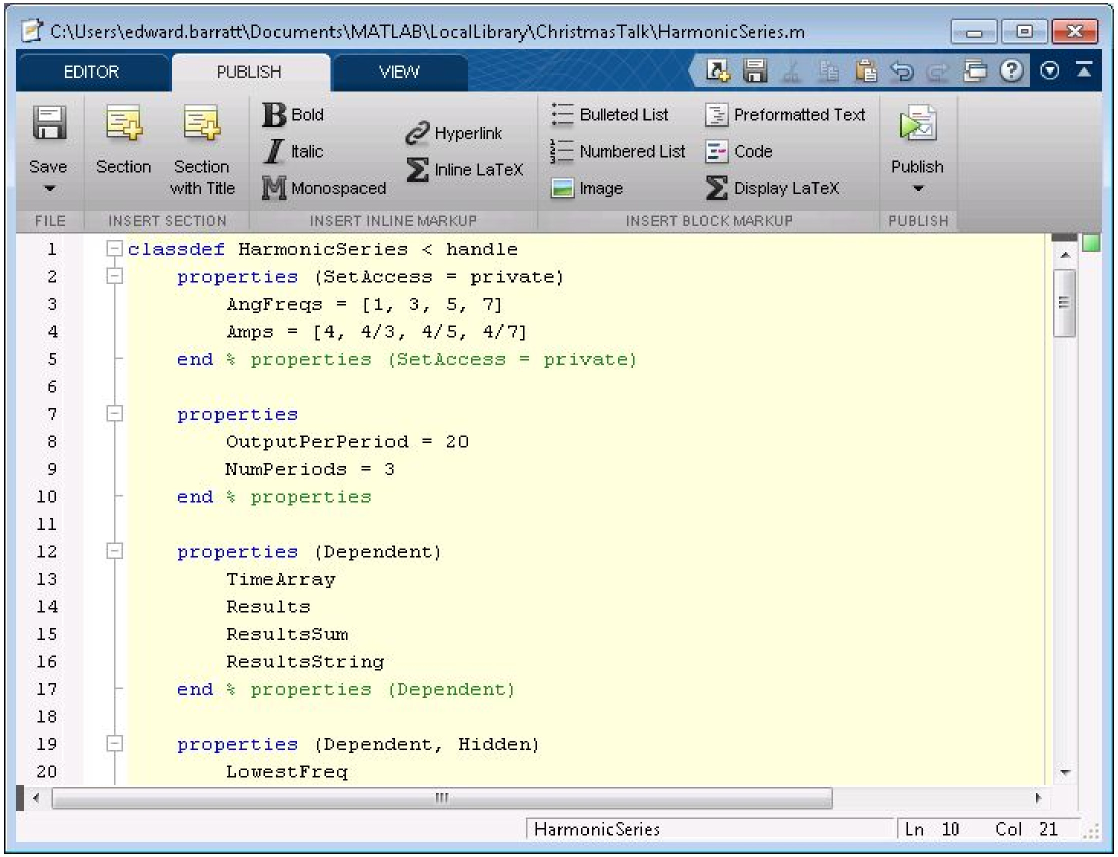
U (m/s) 0.8
Direction 64
00:00 Sun 22nd April 2012
Year 2012
Month April
Day 22
Hour 00
PM10
Wellington Road 5m
Sample Value 21.10 at 394400 804783
Modelled Limits 17.99 to 25.63
Plot Limits 17 to 26 Auto? ☑

Scottish Environment Protection Agency

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

# Object Oriented Programming

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



C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeries.m

```
classdef HarmonicSeries < handle
    properties (SetAccess = private)
        AngFreqs = [1, 3, 5, 7]
        Amps = [4, 4/3, 4/5, 4/7]
    end % properties (SetAccess = private)

    properties
        OutputPerPeriod = 20
        NumPeriods = 3
    end % properties

    properties (Dependent)
        TimeArray
        Results
        ResultsSum
        ResultsString
    end % properties (Dependent)

    properties (Dependent, Hidden)
        LowestFreq
```

SEPA
Scottish Environment Protection Agency

# Object Oriented Programming

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

## 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)'
```

SEPA
Scottish Environment
Protection Agency

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

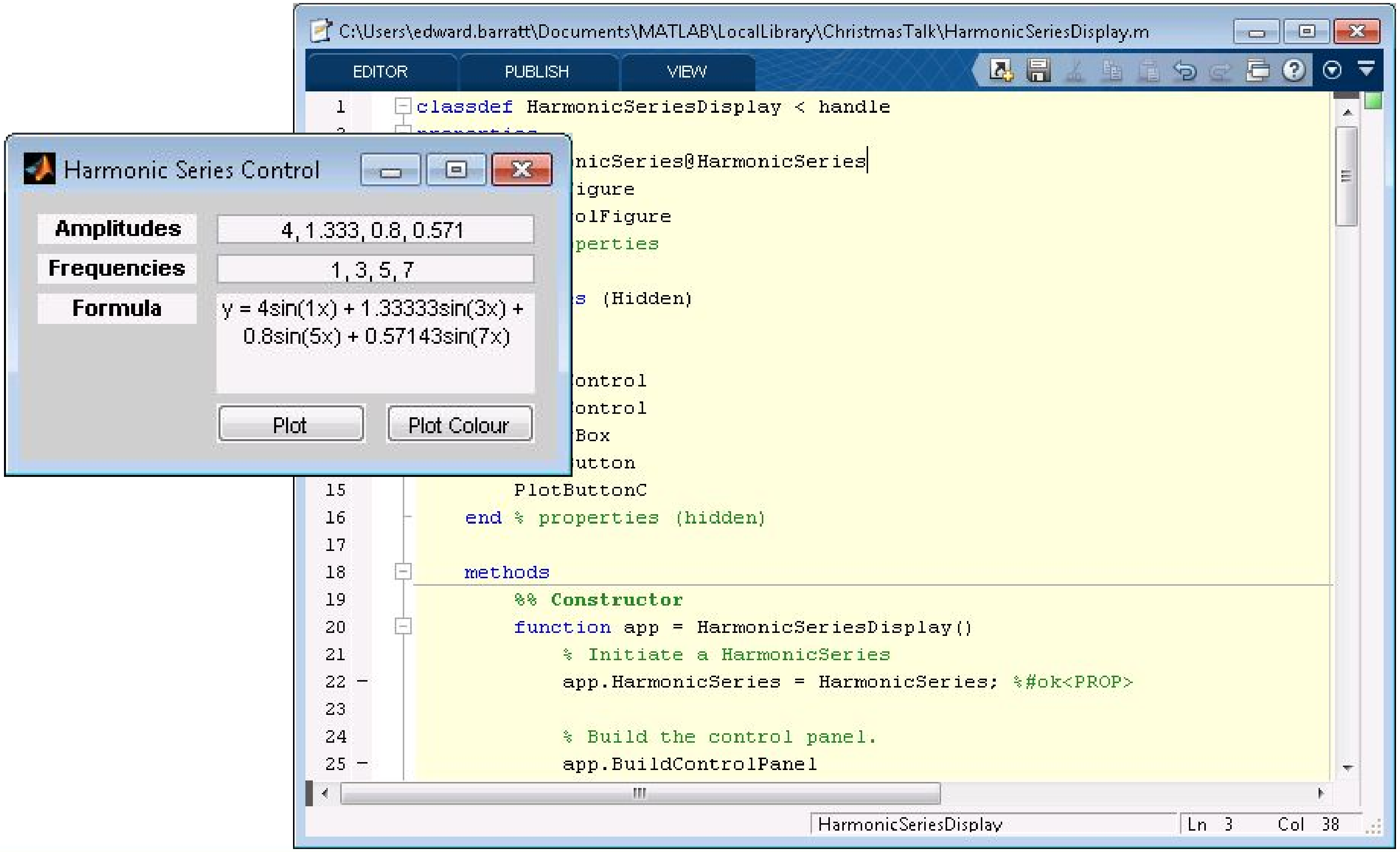
# Object Oriented Programming

- 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.



C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeriesDisplay.m
EDITOR
PUBLISH
VIEW
HarmonicSeriesDisplay
Ln 3
Col 38
Harmonic Series Control
Plot
Plot Colour

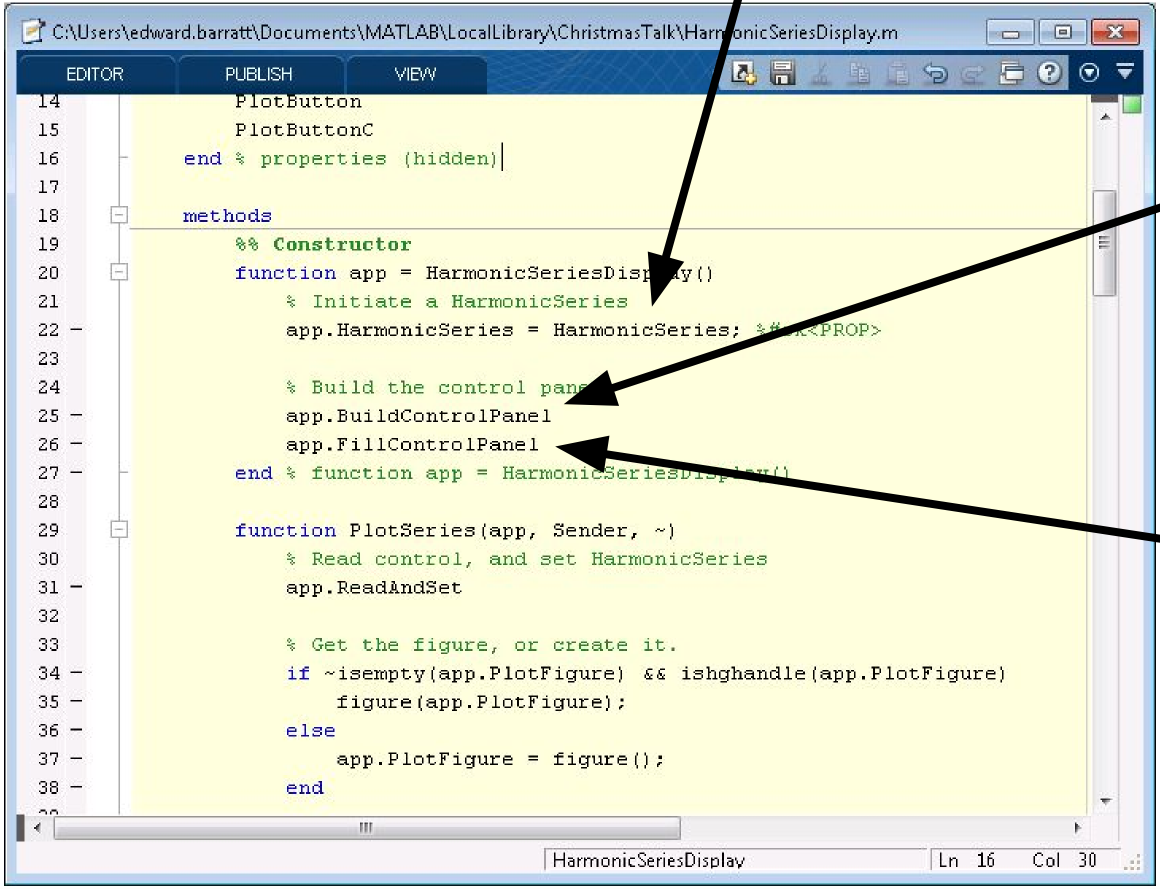
| | |
|---|---|
| Amplitudes | 4, 1.333, 0.8, 0.571 |
| Frequencies | 1, 3, 5, 7 |
| Formula | y = 4sin(1x) + 1.33333sin(3x) + 0.8sin(5x) + 0.57143sin(7x) |

```
classdef HarmonicSeriesDisplay < handle
    properties
              nicSeries@HarmonicSeries
              igure
              olFigure
              perties

              s (Hidden)


              ontrol
              ontrol
              Box
              utton
            PlotButtonC
    end % properties (hidden)

    methods
        %% Constructor
        function app = HarmonicSeriesDisplay()
            % Initiate a HarmonicSeries
            app.HarmonicSeries = HarmonicSeries; %#ok<PROP>

            % Build the control panel.
            app.BuildControlPanel
```

# A GUI Class

## Constructor

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


C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeriesDisplay.m
EDITOR
PUBLISH
VIEW
HarmonicSeriesDisplay
Ln 16
Col 30

```
        PlotButton
        PlotButtonC
    end % properties (hidden)

    methods
        %% Constructor
        function app = HarmonicSeriesDisplay()
            % Initiate a HarmonicSeries
            app.HarmonicSeries = HarmonicSeries; %#ok<PROP>

            % Build the control panel
            app.BuildControlPanel
            app.FillControlPanel
        end % function app = HarmonicSeriesDisplay()

        function PlotSeries(app, Sender, ~)
            % Read control, and set HarmonicSeries
            app.ReadAndSet

            % Get the figure, or create it.
            if ~isempty(app.PlotFigure) && ishghandle(app.PlotFigure)
                figure(app.PlotFigure);
            else
                app.PlotFigure = figure();
            end
```

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

Then it calls the “BuildControlPanel” method

Followed by the “FillControlPanel” method

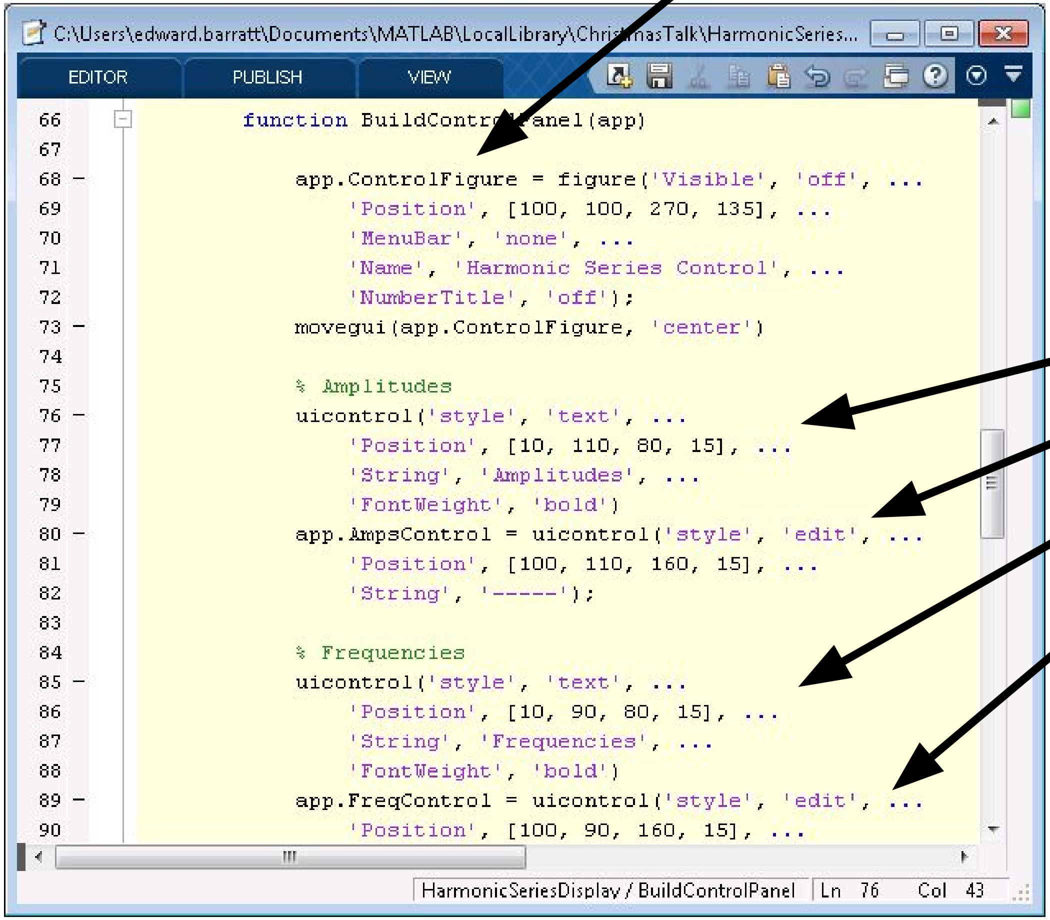
SEPA
Scottish Environment
Protection Agency

- 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.


C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeries...
EDITOR
PUBLISH
VIEW
HarmonicSeriesDisplay / BuildControlPanel
Ln 76
Col 43

```
function BuildControlPanel(app)

    app.ControlFigure = figure('Visible', 'off', ...
        'Position', [100, 100, 270, 135], ...
        'MenuBar', 'none', ...
        'Name', 'Harmonic Series Control', ...
        'NumberTitle', 'off');
    movegui(app.ControlFigure, 'center')

    % Amplitudes
    uicontrol('style', 'text', ...
        'Position', [10, 110, 80, 15], ...
        'String', 'Amplitudes', ...
        'FontWeight', 'bold')
    app.AmpsControl = uicontrol('style', 'edit', ...
        'Position', [100, 110, 160, 15], ...
        'String', '-----');

    % Frequencies
    uicontrol('style', 'text', ...
        'Position', [10, 90, 80, 15], ...
        'String', 'Frequencies', ...
        'FontWeight', 'bold')
    app.FreqControl = uicontrol('style', 'edit', ...
        'Position', [100, 90, 160, 15], ...
```

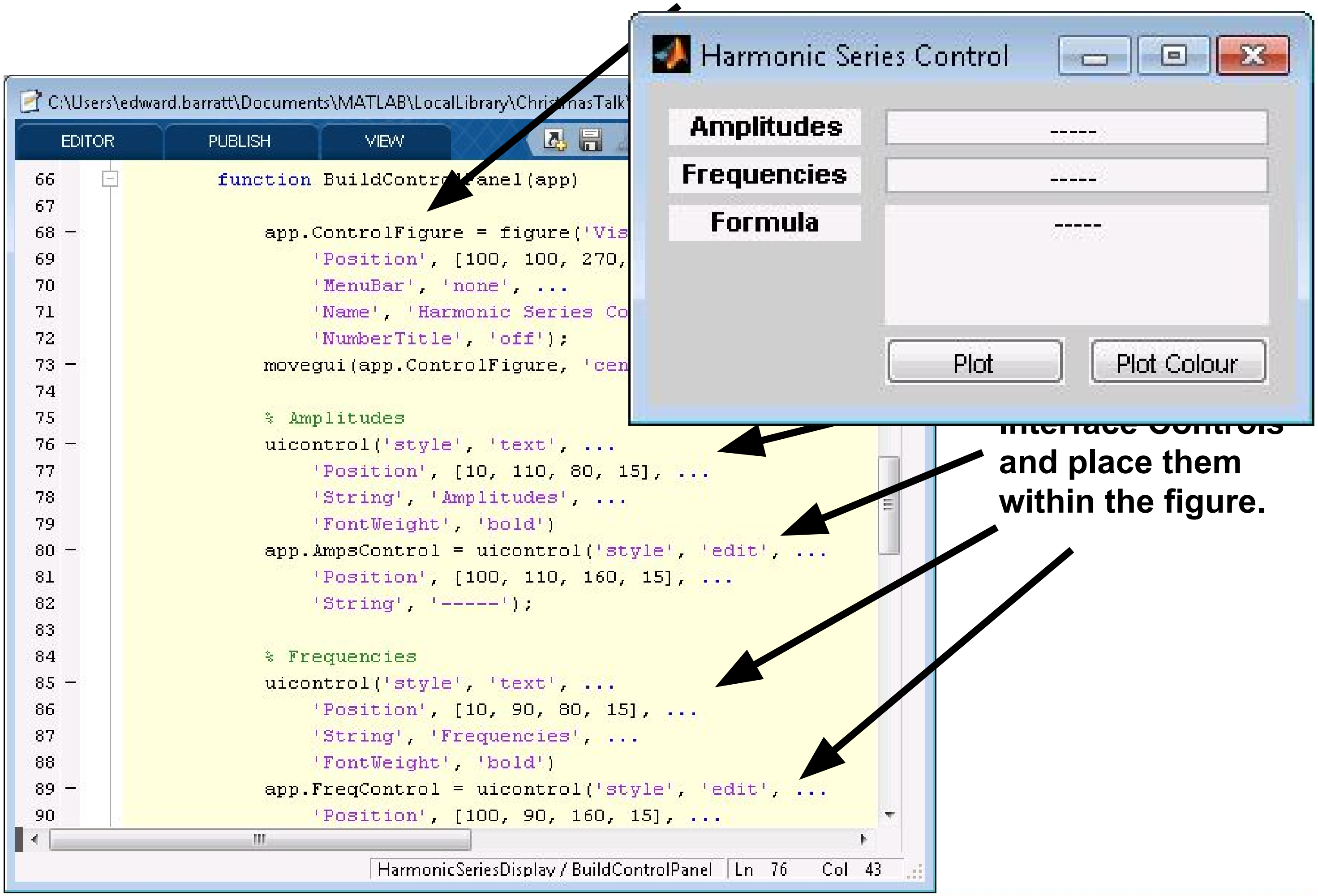
Define various User Interface Controls and place them within the figure.

# A GUI Class - uicontrol

## BuildControlPanel method

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

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


C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\
EDITOR
PUBLISH
VIEW
HarmonicSeriesDisplay / BuildControlPanel
Ln 76
Col 43
Harmonic Series Control
Amplitudes
-----
Frequencies
-----
Formula
-----
Plot
Plot Colour

```
        function BuildControlPanel(app)

            app.ControlFigure = figure('Vis
                'Position', [100, 100, 270,
                'MenuBar', 'none', ...
                'Name', 'Harmonic Series Co
                'NumberTitle', 'off');
            movegui(app.ControlFigure, 'cen

            % Amplitudes
            uicontrol('style', 'text', ...
                'Position', [10, 110, 80, 15], ...
                'String', 'Amplitudes', ...
                'FontWeight', 'bold')
            app.AmpsControl = uicontrol('style', 'edit', ...
                'Position', [100, 110, 160, 15], ...
                'String', '-----');

            % Frequencies
            uicontrol('style', 'text', ...
                'Position', [10, 90, 80, 15], ...
                'String', 'Frequencies', ...
                'FontWeight', 'bold')
            app.FreqControl = uicontrol('style', 'edit', ...
                'Position', [100, 90, 160, 15], ...
```

Interface Controls and place them within the figure.

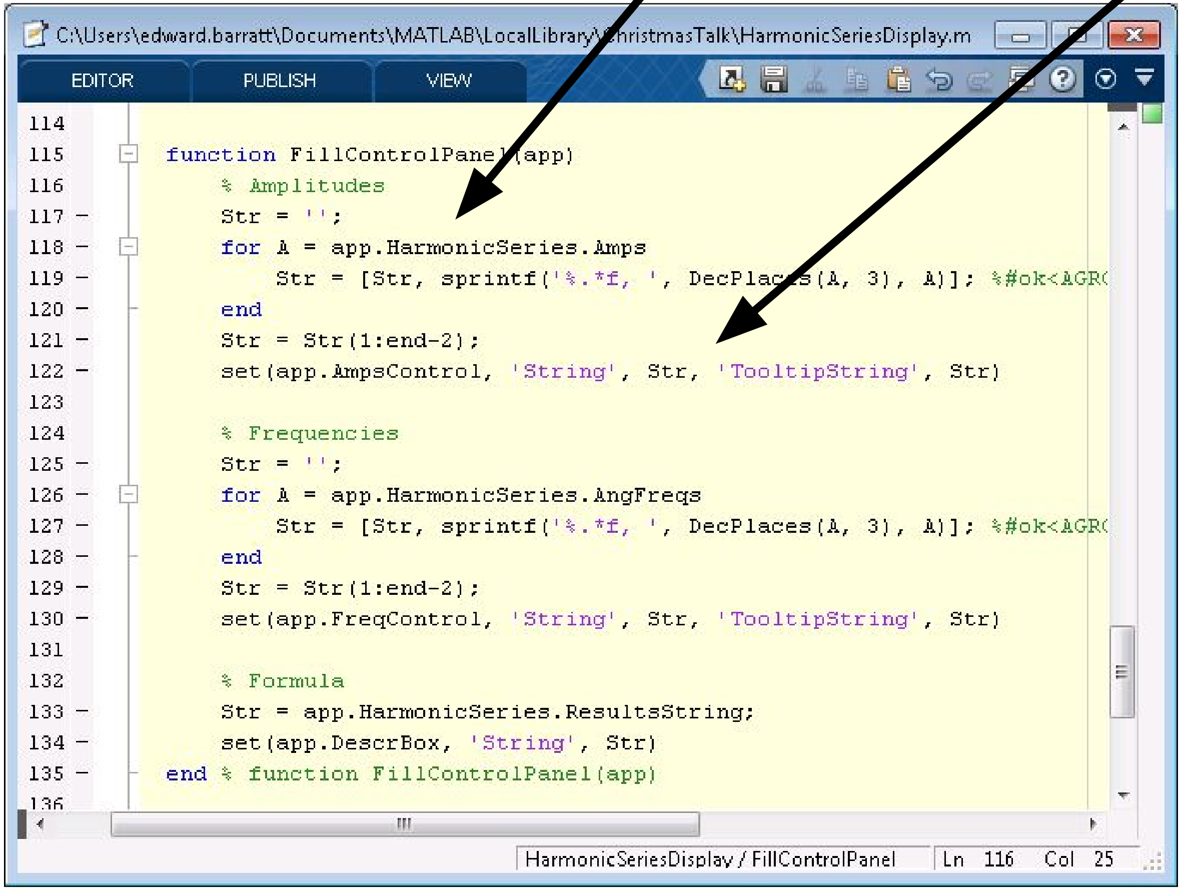
# A GUI Class - uicontrol

## FillControlPanel method

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

Use the values from the HarmonicSeries object to define strings.

And set the ‘String’ properties of the uicontrol objects.


C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeriesDisplay.m
EDITOR
PUBLISH
VIEW
HarmonicSeriesDisplay / FillControlPanel
Ln 116
Col 25

```matlab

function FillControlPanel(app)
    % Amplitudes
    Str = '';
    for A = app.HarmonicSeries.Amps
        Str = [Str, sprintf('%.*f, ', DecPlaces(A, 3), A)]; %#ok<AGR
    end
    Str = Str(1:end-2);
    set(app.AmpsControl, 'String', Str, 'TooltipString', Str)

    % Frequencies
    Str = '';
    for A = app.HarmonicSeries.AngFreqs
        Str = [Str, sprintf('%.*f, ', DecPlaces(A, 3), A)]; %#ok<AGR
    end
    Str = Str(1:end-2);
    set(app.FreqControl, 'String', Str, 'TooltipString', Str)

    % Formula
    Str = app.HarmonicSeries.ResultsString;
    set(app.DescrBox, 'String', Str)
end % function FillControlPanel(app)
```

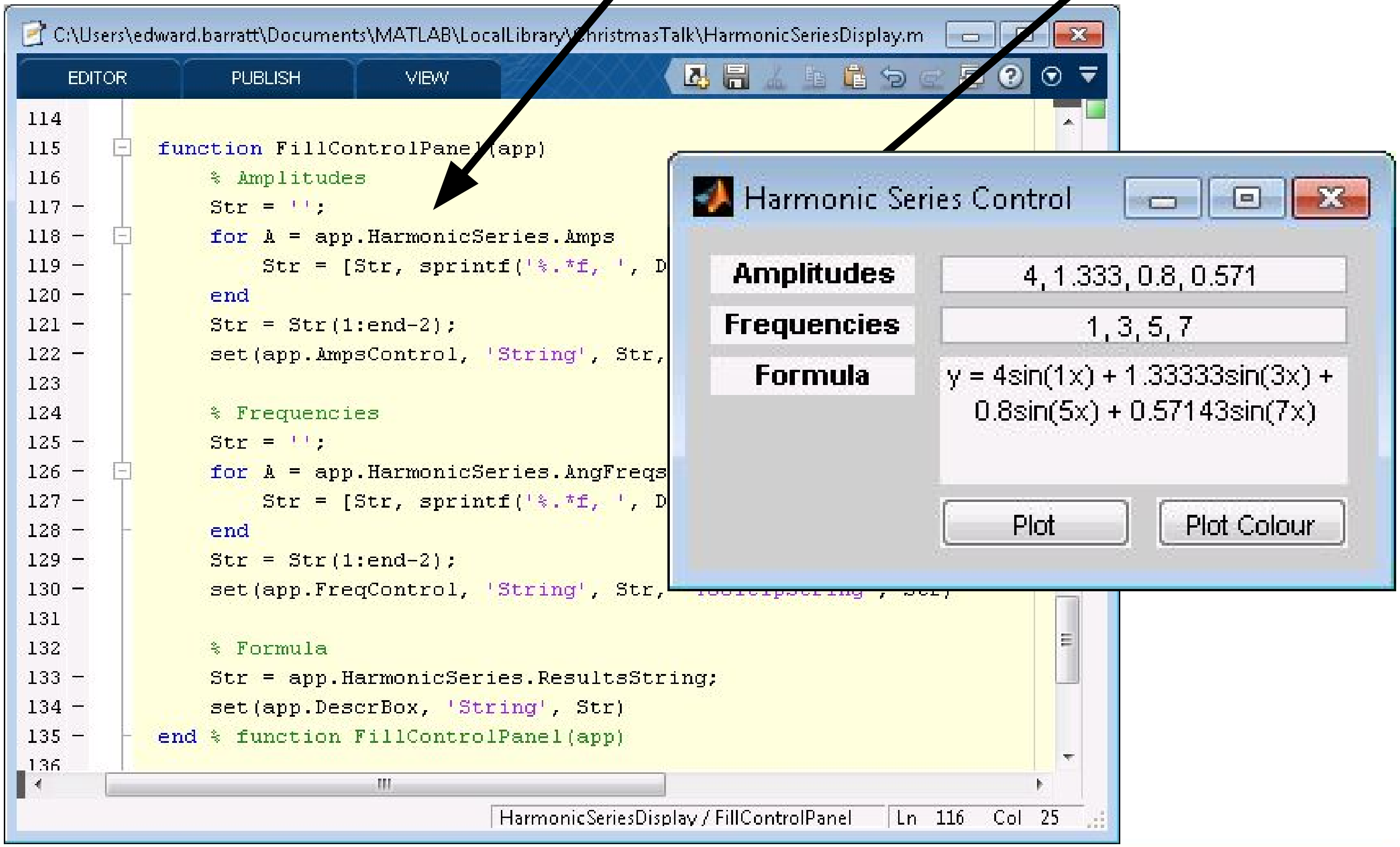
# A GUI Class - uicontrol

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

## FillControlPanel method

**Use the values from the HarmonicSeries object to define strings.**

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



C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeriesDisplay.m

```
function FillControlPanel(app)
    % Amplitudes
    Str = '';
    for A = app.HarmonicSeries.Amps
        Str = [Str, sprintf('%.*f, ', D
    end
    Str = Str(1:end-2);
    set(app.AmpsControl, 'String', Str,

    % Frequencies
    Str = '';
    for A = app.HarmonicSeries.AngFreqs
        Str = [Str, sprintf('%.*f, ', D
    end
    Str = Str(1:end-2);
    set(app.FreqControl, 'String', Str, 'TooltipString', Str);

    % Formula
    Str = app.HarmonicSeries.ResultsString;
    set(app.DescrBox, 'String', Str)
end % function FillControlPanel(app)
```

HarmonicSeriesDisplay / FillControlPanel    Ln 116    Col 25

Harmonic Series Control

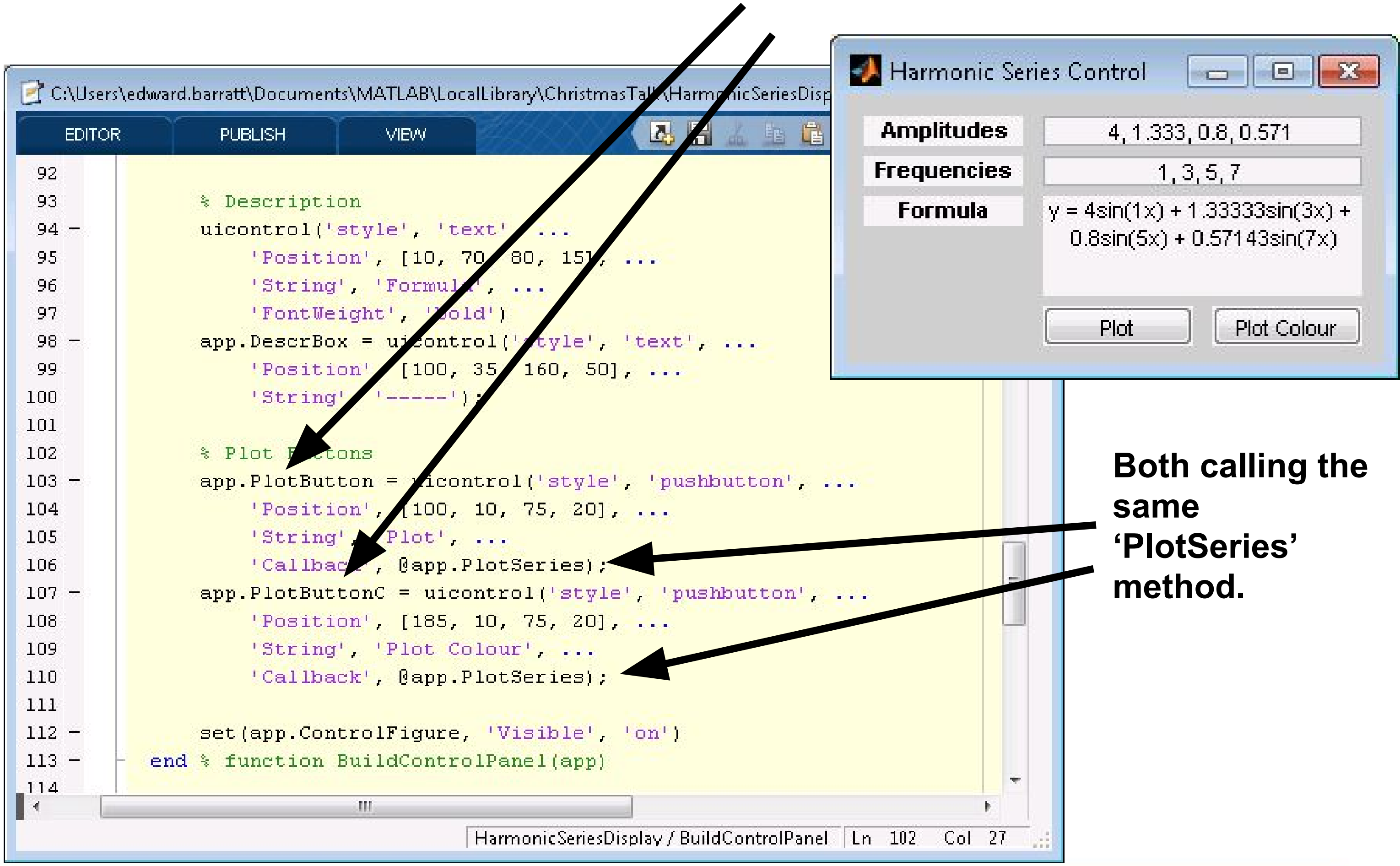
| | |
|---|---|
| **Amplitudes** | 4, 1.333, 0.8, 0.571 |
| **Frequencies** | 1, 3, 5, 7 |
| **Formula** | y = 4sin(1x) + 1.33333sin(3x) + 0.8sin(5x) + 0.57143sin(7x) |

# A GUI Class - callbacks

## BuildControlPanel method

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

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


C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeriesDisp
EDITOR
PUBLISH
VIEW
Harmonic Series Control
Plot
Plot Colour

```
    % Description
    uicontrol('style', 'text', ...
        'Position', [10, 70, 80, 15], ...
        'String', 'Formula', ...
        'FontWeight', 'bold')
    app.DescrBox = uicontrol('style', 'text', ...
        'Position', [100, 35, 160, 50], ...
        'String', '-----');

    % Plot Buttons
    app.PlotButton = uicontrol('style', 'pushbutton', ...
        'Position', [100, 10, 75, 20], ...
        'String', 'Plot', ...
        'Callback', @app.PlotSeries);
    app.PlotButtonC = uicontrol('style', 'pushbutton', ...
        'Position', [185, 10, 75, 20], ...
        'String', 'Plot Colour', ...
        'Callback', @app.PlotSeries);

    set(app.ControlFigure, 'Visible', 'on')
end % function BuildControlPanel(app)
```

HarmonicSeriesDisplay / BuildControlPanel   Ln 102   Col 27

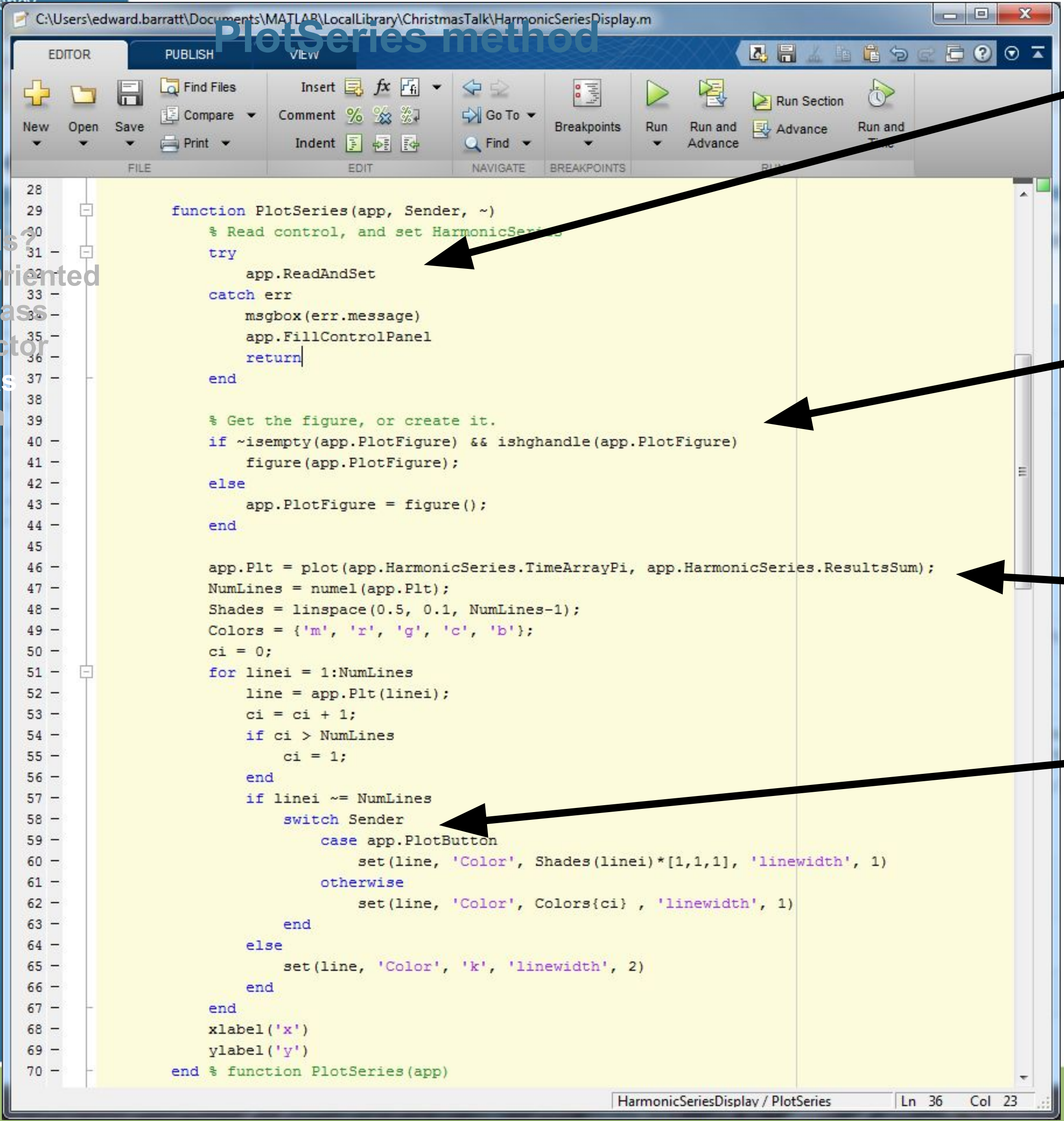
| | |
|---|---|
| **Amplitudes** | 4, 1.333, 0.8, 0.571 |
| **Frequencies** | 1, 3, 5, 7 |
| **Formula** | y = 4sin(1x) + 1.33333sin(3x) + 0.8sin(5x) + 0.57143sin(7x) |

**Both calling the same 'PlotSeries' method.**

# A GUI Class - callbacks

## PlotSeries method

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



C:\Users\edward.barratt\Documents\MATLAB\LocalLibrary\ChristmasTalk\HarmonicSeriesDisplay.m

```matlab

    function PlotSeries(app, Sender, ~)
        % Read control, and set HarmonicSeries
        try
            app.ReadAndSet
        catch err
            msgbox(err.message)
            app.FillControlPanel
            return
        end

        % Get the figure, or create it.
        if ~isempty(app.PlotFigure) && ishghandle(app.PlotFigure)
            figure(app.PlotFigure);
        else
            app.PlotFigure = figure();
        end

        app.Plt = plot(app.HarmonicSeries.TimeArrayPi, app.HarmonicSeries.ResultsSum);
        NumLines = numel(app.Plt);
        Shades = linspace(0.5, 0.1, NumLines-1);
        Colors = {'m', 'r', 'g', 'c', 'b'};
        ci = 0;
        for linei = 1:NumLines
            line = app.Plt(linei);
            ci = ci + 1;
            if ci > NumLines
                ci = 1;
            end
            if linei ~= NumLines
                switch Sender
                    case app.PlotButton
                        set(line, 'Color', Shades(linei)*[1,1,1], 'linewidth', 1)
                    otherwise
                        set(line, 'Color', Colors{ci} , 'linewidth', 1)
                end
            else
                set(line, 'Color', 'k', 'linewidth', 2)
            end
        end
        xlabel('x')
        ylabel('y')
    end % function PlotSeries(app)
```

HarmonicSeriesDisplay / PlotSeries    Ln 36    Col 23

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

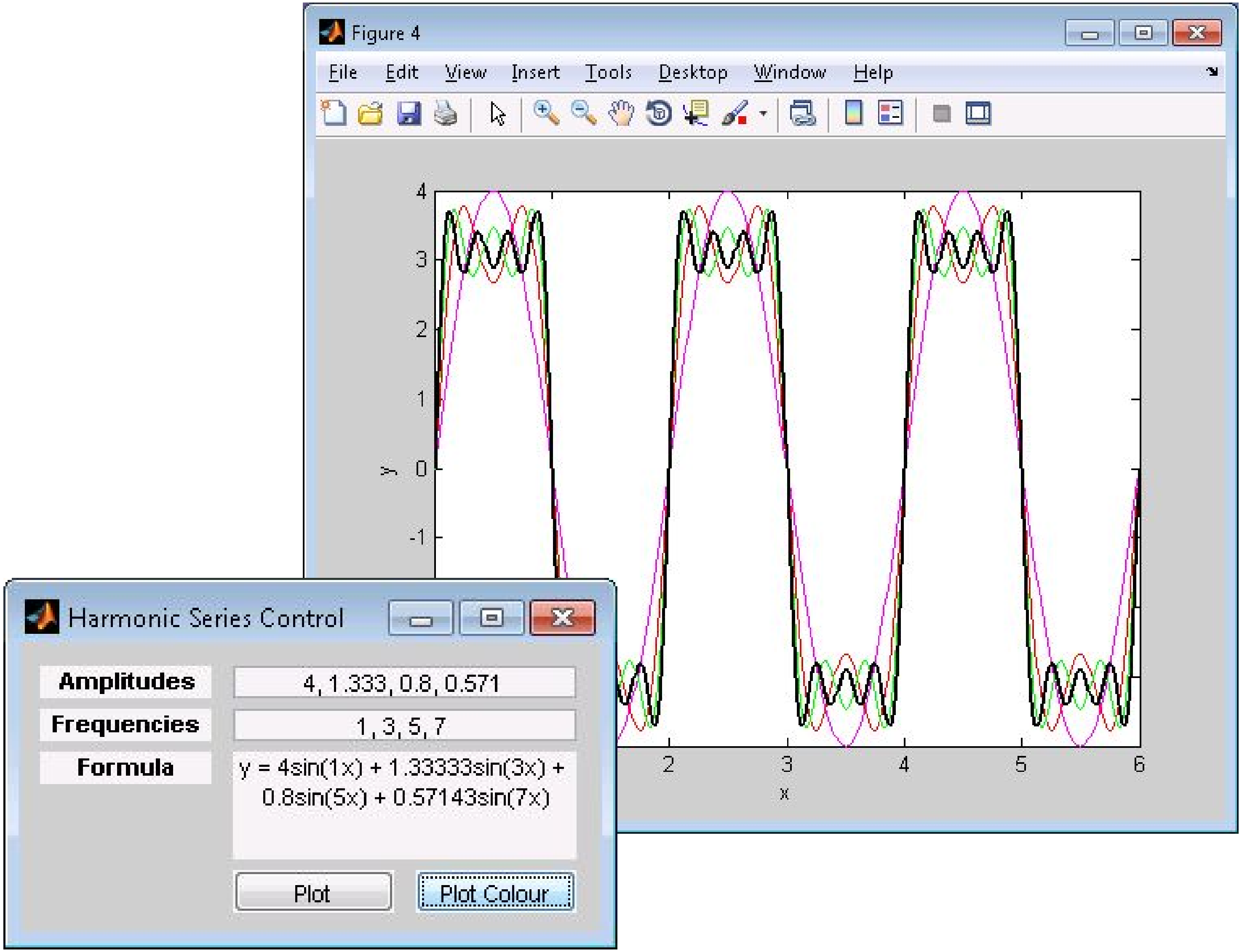
Open a figure, or get it if it’s already been opened.

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



Amplitudes: 4, 1.333, 0.8, 0.571

Frequencies: 1, 3, 5, 7

Formula: y = 4sin(1x) + 1.33333sin(3x) + 0.8sin(5x) + 0.57143sin(7x)