

Learning Python

Getting results for beamlines and scientific programming

Intermediate python: GUI programming with WxPython, part 4

Outline of topics to be covered

- Fixed space in sizers
- Implementing drag and drop



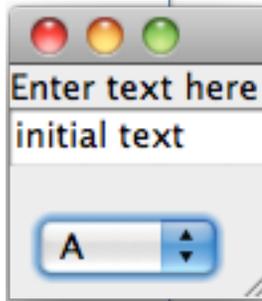
Adding a fixed space to a sizer

```
def __init__(self):
    wx.Frame.__init__(self, None)
    pnl = wx.Panel(self)
    sizer = wx.BoxSizer(wx.VERTICAL)
    pnl.SetSizer(sizer)

    sizer.Add(wx.StaticText(pnl,-1,
        'Enter text here'))
    bx = wx.TextCtrl(pnl, -1,
        "initial text",
        size=(100,-1))
    sizer.Add(bx,0,wx.BOTTOM,10)

    lbls = ['A','B','C']
    rbx = wx.Choice(pnl,-1,
        choices=lbls)
    sizer.Add(rbx,0,wx.ALL,10)

    sizer.Fit(self)
    self.Show()
```



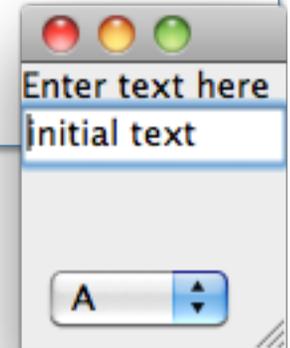
```
def __init__(self):
    wx.Frame.__init__(self, None)
    pnl = wx.Panel(self)
    sizer = wx.BoxSizer(wx.VERTICAL)
    pnl.SetSizer(sizer)

    sizer.Add(wx.StaticText(pnl,-1,
        'Enter text here'))
    bx = wx.TextCtrl(pnl, -1,
        "initial text",
        size=(100,-1))
    sizer.Add(bx)

    sizer.Add(wx.Size(-1,30))

    lbls = ['A','B','C']
    rbx = wx.Choice(pnl,-1,
        choices=lbls)
    sizer.Add(rbx,0,wx.ALL,10)

    sizer.Fit(self)
    self.Show()
```



A good way to put space between widgets is to insert a blank space of a fixed size. Here the object has no length but is 30 points high (intentionally exaggerated).

Drag and Drop

First define an event handler to be called when an object is dragged

- This defines a “payload” to be sent (here this is text taken from the label)
- Then define an object that is associated with the drag operation (drpTxtObj)

Then bind the objects to be dropped

```
def OnDragLbl(event):
    'Respond to a drag of a Label'
    w = event.GetEventObject()
    txt = w.GetLabelText()
    drpTxtMsg = wx.PyTextDataObject(txt)
    drpTxtObj = wx.DropSource(w)
    drpTxtObj.SetData(drpTxtMsg)
    # Intiate the Drag Operation
    drpTxtObj.DoDragDrop(True)
```

```
def green(pSizer, panel):
    '''construct a 3rd level sizer'''
    sizer = wx.BoxSizer(wx.VERTICAL)
    for t in ['1', '2', '3']:
        txt = wx.StaticText(panel, -1, t)
        txt.SetBackgroundColour('gray')
        txt.SetForegroundColour('green')
        sizer.Add(txt, 0, wx.ALL, 10)
        # bind a drag command to the label
        txt.Bind(wx.EVT_LEFT_DOWN,
                OnDragLbl)
    return sizer
```



Drop Target Infrastructure

Define a class that saves information about the target (where the item will be dropped) and the action to be performed when the drop occurs

- Here we save a reference to the target and an optional action to perform
- As a demo we change the label in the target to the text in the payload

```
# Define a class for dropping text
class DropTarget(wx.TextDropTarget):
    """ This object implements Drop Target functionality for Text """
    def __init__(self, DropTarget, action=None):
        """ Initialize the Drop Target, passing in a list of objects to
            process the dropped text """
        # Initialize the Target Object
        wx.TextDropTarget.__init__(self)
        # Store the Object Reference for dropped text
        self.DropTarget = DropTarget
        self.DropTargetAction = action

    def OnDropText(self, x, y, data):
        """ Implement Text Drop """
        # When text is dropped, process it with the specified commands
        if self.DropTargetAction != None:
            self.DropTargetAction
            print data, 'dropped on', self.DropTarget
            self.DropTarget.SetLabel(data)
```

dnd.py

Set up target

Make a drop object for the target widget (might have an optional action in call)

Use the widget's SetDropTarget method to associate that with the widget

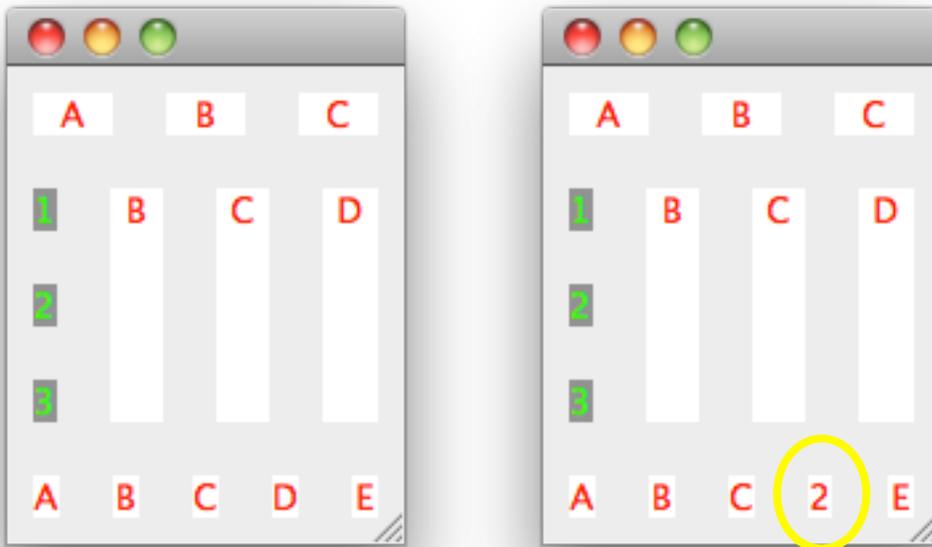
```
def red(pSizer, panel, num):
    '''construct a 2nd level sizer.
    Behaves differently by num'''
    sizer = wx.BoxSizer(wx.HORIZONTAL)
    if num == 1:
        lbls = ['A', 'B', 'C']
    elif num == 2:
        lbls = ['B', 'C', 'D']
        greensizer = green(sizer, panel)
        sizer.Add(greensizer, 0, wx.ALL, 0)
    else:
        lbls = ['A', 'B', 'C', 'D', 'E']
    for t in lbls:
        txt = wx.StaticText(panel, wx.ID_ANY,
                             t, style=wx.ALIGN_CENTER)
        txt.SetBackgroundColour('white')
        txt.SetForegroundColour('red')
        sizer.Add(txt, 1, wx.ALL|wx.EXPAND, 10)
        target = DropTarget(txt)
        txt.SetDropTarget(target)
    return sizer
```

dnd.py



Drag & Drop

Result from dragging the “2” to the “D”



Homework

- Create a small wxPython program to do something useful. Let me know what you did or where you get stuck.

