

Einbau wxMenu

Einbau wxMenu

Einbau wxMenu

```
Kurs-Version-1.py - /home/nutzer/Dokumente/LI/00/2023-LI-00/Projekte/07/Kurs-Version-2023/Kurs-Version-1.py (3.6.15) x
File Edit Format Run Options Window Help
# Kurs-Version-1.py , basiert auf # 0_leeres_Fenster.py

import wx
#-----
class MyFrame(wx.Frame):
    def __init__(
        self, parent, ID, title, pos=wx.DefaultPosition,
        size=wx.DefaultSize, style=wx.DEFAULT_FRAME_STYLE
    ):

        wx.Frame.__init__(self, parent, ID, title, pos, size, style)
        panel = wx.Panel(self, -1)

        button = wx.Button(panel, 1003, "Close Me")
        button.SetPosition((15, 15))
        self.Bind(wx.EVT_BUTTON, self.OnCloseMe, button)
        self.Bind(wx.EVT_CLOSE, self.OnCloseWindow)

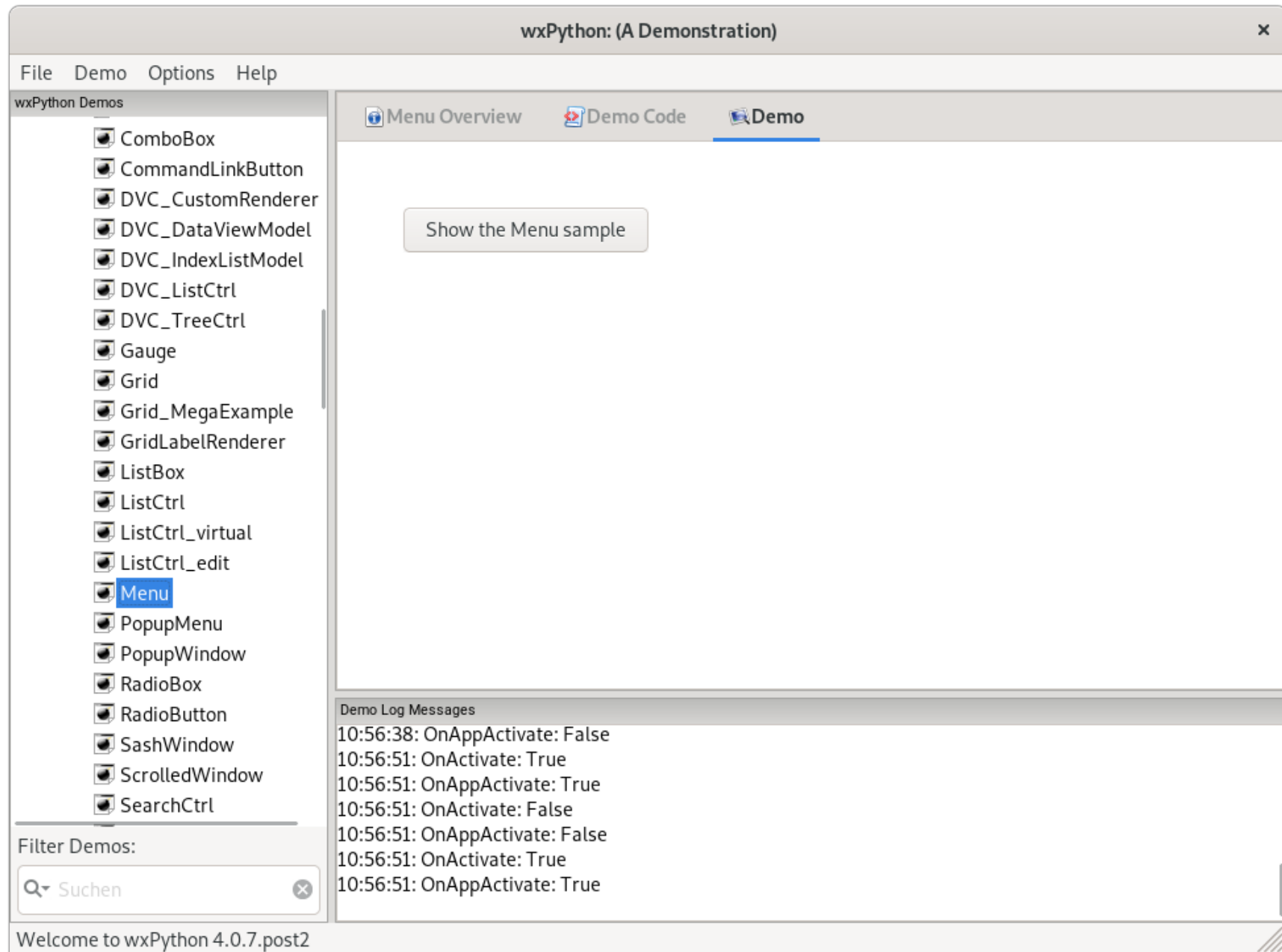
    def OnCloseMe(self, event):
        self.Close(True)

    def OnCloseWindow(self, event):
        self.Destroy()

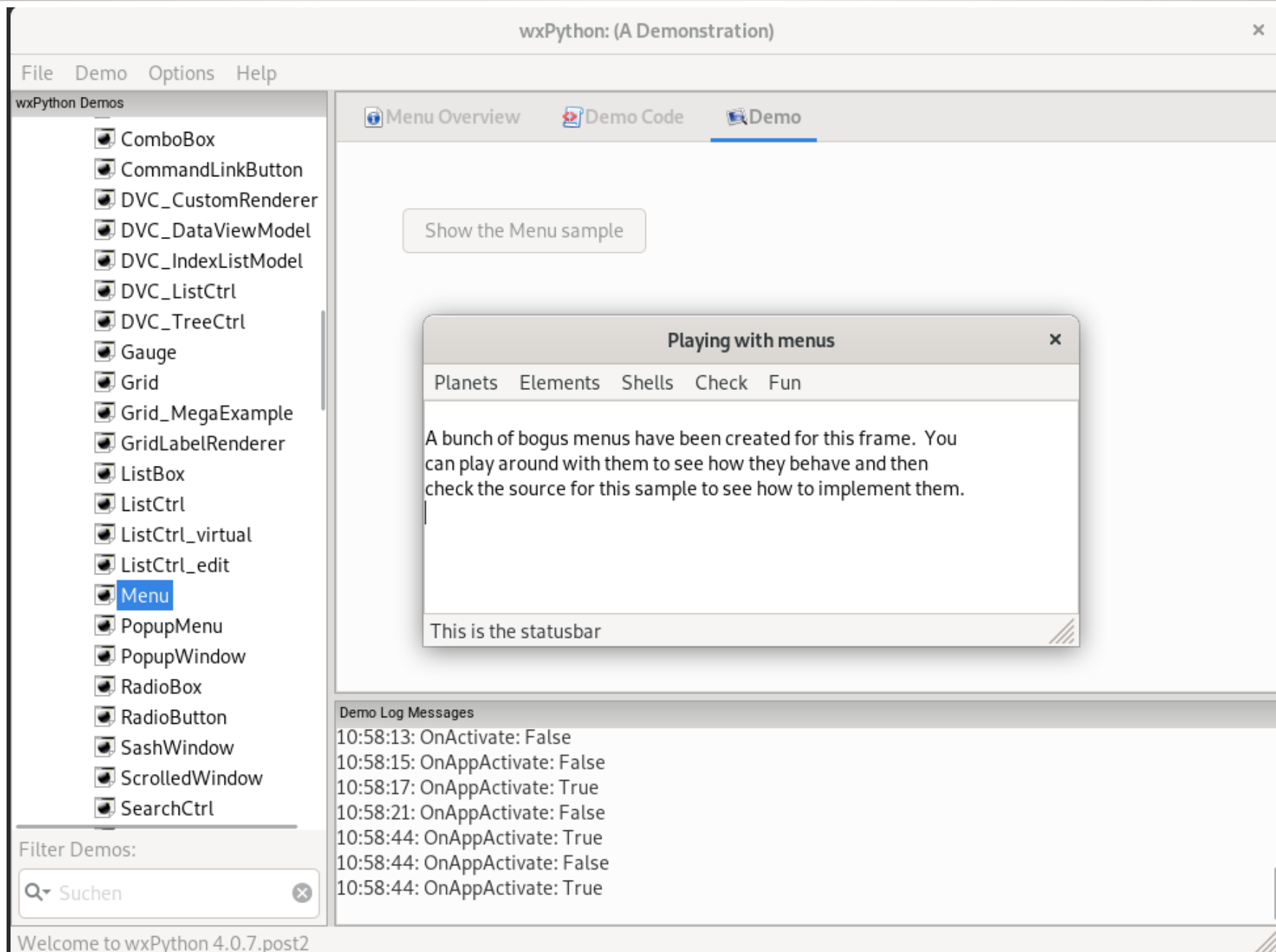
#-----
if __name__=='__main__':
    app=wx.App()
    fenster=MyFrame(None, -1, "Fenster")
    app.SetTopWindow(fenster)
    fenster.Show(True)
    app.MainLoop()
```

Ln: 1 Col: 37

Einbau wxMenu



Einbau wxMenu



Einbau wxMenu

The screenshot shows the wxPython: (A Demonstration) application window. The interface includes a menu bar with 'File', 'Demo', 'Options', and 'Help'. On the left, a tree view under 'wxPython Demos' lists various widgets, with 'Menu' selected. The main area is divided into three tabs: 'Menu Overview', 'Demo Code', and 'Demo'. The 'Demo Code' tab is active, showing Python code for a class named 'MyFrame'. The code includes imports for 'time', 'wx', and 'images', and defines an '__init__' method that sets up a frame with a status bar and a text control. The status bar text is 'This is the statusbar'. The text control contains a message about bogus menus. At the bottom, a 'Demo Log Messages' panel shows a series of log entries. A search bar at the bottom left contains the text 'Suchen'.

```
1  #!/usr/bin/env python
2
3  import time
4  import wx
5  import images
6
7  #-----
8
9  class MyFrame(wx.Frame):
10
11      def __init__(self, parent, id, log):
12          wx.Frame.__init__(self, parent, id, 'Playing with menus', size=(500, 300))
13          self.log = log
14          self.CenterOnScreen()
15
16          self.CreateStatusBar()
17          self.SetStatusText("This is the statusbar")
18
19          tc = wx.TextCtrl(self, -1, "A bunch of bogus menus have been created for this frame. You can play around with them to see how they behave and then")
20
21
```

Active Version: ☒ Original ☐ Modified

10:59:45: OnAppActivate: False
10:59:57: OnActivate: True
10:59:57: OnAppActivate: True
10:59:57: OnActivate: False
10:59:57: OnAppActivate: False
10:59:57: OnActivate: True
10:59:57: OnAppActivate: True

Welcome to wxPython 4.0.7.post2

Einbau wxMenu

The screenshot shows the 'wxPython: (A Demonstration)' application window. The interface includes a menu bar with 'File', 'Demo', 'Options', and 'Help'. On the left, a 'wxPython Demos' list contains various widget examples, with 'Menu' selected. The main area is divided into three tabs: 'Menu Overview', 'Demo Code', and 'Demo'. The 'Demo Code' tab is active, displaying Python code for creating a menu. The code defines a menu bar, a first menu with items like '&Mercury', '&Venus', '&Earth', and '&Close', and a second menu with '&Hydrogen' and '&Helium'. A submenu is also indicated for the second menu. Below the code, a 'Demo Log Messages' panel shows a series of log entries related to application activation. At the bottom, a status bar displays 'Welcome to wxPython 4.0.7.post2'.

wxPython: (A Demonstration)

File Demo Options Help

wxPython Demos

- ComboBox
- CommandLinkButton
- DVC_CustomRenderer
- DVC_DataViewModel
- DVC_IndexListModel
- DVC_ListCtrl
- DVC_TreeCtrl
- Gauge
- Grid
- Grid_MegaExample
- GridLabelRenderer
- ListBox
- ListCtrl
- ListCtrl_virtual
- ListCtrl_edit
- Menu**
- PopupMenu
- PopupWindow
- RadioBox
- RadioButton
- SashWindow
- ScrolledWindow
- SearchCtrl

Filter Demos:

Suchen

Menu Overview Demo Code Demo

Active Version: ☒ Original ☐ Modified Save Changes Delete Modified

```
22 check the source for this sample to see how to implement them.
23 """ , style=wx.TE_READONLY|wx.TE_MULTILINE)
24
25 # Prepare the menu bar
26 menuBar = wx.MenuBar()
27
28 # 1st menu from left
29 menu1 = wx.Menu()
30 menu1.Append(101, "&Mercury", "This the text in the Statusbar")
31 menu1.Append(102, "&Venus", "")
32 menu1.Append(103, "&Earth", "You may select Earth too")
33 menu1.AppendSeparator()
34 menu1.Append(104, "&Close", "Close this frame")
35 # Add menu to the menu bar
36 menuBar.Append(menu1, "&Planets")
37
38 # 2nd menu from left
39 menu2 = wx.Menu()
40 menu2.Append(201, "Hydrogen")
41 menu2.Append(202, "Helium")
42 # a submenu in the 2nd menu
```

Demo Log Messages

11:00:00: OnAppActivate: False
11:00:36: OnActivate: True
11:00:36: OnAppActivate: True
11:00:36: OnActivate: False
11:00:36: OnAppActivate: False
11:00:36: OnActivate: True
11:00:36: OnAppActivate: True

Welcome to wxPython 4.0.7.post2

Einbau wxMenu

The screenshot displays the wxPython IDE interface. On the left, a 'wxPython Demos' list includes various widgets, with 'Menu' selected. The 'Menu Overview' pane shows a tree structure of menu items. The main editor window displays the code for 'Kurs-Version-2.py', which is based on '0_leeres_Fenster.py'. The code defines a 'MyFrame' class that inherits from 'wx.Frame'. It includes a 'def __init__' method that sets up the frame's parent, ID, title, position, size, and style. It also creates a 'panel' and a 'button' with the text 'Close Me'. The 'button' is positioned at (15, 15) and is bound to the 'OnCloseMe' event. The 'OnCloseMe' method calls 'self.Close(True)'. Additionally, the code prepares a menu bar and adds a menu named '&Planets' with several items: '&Mercury', '&Venus', '&Earth', a separator, and '&Close'. The 'OnCloseWindow' method calls 'self.Destroy()'. The 'Demo Log Messages' pane at the bottom shows a series of log messages related to the application's activation and deactivation.

```
File Edit Format Run Options Window Help
# Kurs-Version-2.py , basiert auf # 0_leeres_Fenster.py

import wx
#-----
class MyFrame(wx.Frame):
    def __init__(
        self, parent, ID, title, pos=wx.DefaultPosition,
        size=wx.DefaultSize, style=wx.DEFAULT_FRAME_STYLE
    ):
        wx.Frame.__init__(self, parent, ID, title, pos, size, style)
        panel = wx.Panel(self, -1)

        button = wx.Button(panel, 1003, "Close Me")
        button.SetPosition((15, 15))
        self.Bind(wx.EVT_BUTTON, self.OnCloseMe, button)
        self.Bind(wx.EVT_CLOSE, self.OnCloseWindow)

        # Prepare the menu bar
        menuBar = wx.MenuBar()

        # 1st menu from left
        menu1 = wx.Menu()
        menu1.Append(101, "&Mercury", "This the text in the Statusbar")
        menu1.Append(102, "&Venus", "")
        menu1.Append(103, "&Earth", "You may select Earth too")
        menu1.AppendSeparator()
        menu1.Append(104, "&Close", "Close this frame")
        # Add menu to the menu bar
        menuBar.Append(menu1, "&Planets")

        def OnCloseMe(self, event):
            self.Close(True)

        def OnCloseWindow(self, event):
            self.Destroy()
```

11:01:25: OnAppActivate: False
11:01:30: OnActivate: True
11:01:30: OnAppActivate: True
11:01:30: OnActivate: False
11:01:30: OnAppActivate: False
11:01:30: OnActivate: True
11:01:30: OnAppActivate: True

Welcome to wxPython 4.0.7.post2

Einbau wxMenu

The screenshot shows the 'wxPython: (A Demonstration)' application window. The 'Menu' demo is selected in the left sidebar. The main area displays the 'Demo Code' tab, showing Python code for setting up a menu. The code includes menu items, event bindings, and a log of messages at the bottom.

wxPython Demos:

- ComboBox
- CommandLinkButton
- DVC_CustomRenderer
- DVC_DataViewModel
- DVC_IndexListModel
- DVC_ListCtrl
- DVC_TreeCtrl
- Gauge
- Grid
- Grid_MegaExample
- GridLabelRenderer
- ListBox
- ListCtrl
- ListCtrl_virtual
- ListCtrl_edit
- Menu**
- PopupMenu
- PopupWindow
- RadioBox
- RadioButton
- SashWindow
- ScrolledWindow
- SearchCtrl

Filter Demos:

Menu Overview | Demo Code | Demo

Active Version: ☒ Original ☐ Modified

```
98 menu5.Append(507, "insert after this...")
99 menu5.Append(508, "...and before this")
100 | menuBar.Append(menu5, "&Fun")
101
102 self.SetMenuBar(menuBar)
103
104 # Menu events
105 self.Bind(wx.EVT_MENU_HIGHLIGHT_ALL, self.OnMenuHighlight)
106
107 self.Bind(wx.EVT_MENU, self.Menu101, id=101)
108 self.Bind(wx.EVT_MENU, self.Menu102, id=102)
109 self.Bind(wx.EVT_MENU, self.Menu103, id=103)
110 self.Bind(wx.EVT_MENU, self.CloseWindow, id=104)
111
112 self.Bind(wx.EVT_MENU, self.Menu201, id=201)
113 self.Bind(wx.EVT_MENU, self.Menu202, id=202)
114 self.Bind(wx.EVT_MENU, self.Menu2031, id=2031)
115 self.Bind(wx.EVT_MENU, self.Menu2032, id=2032)
116 self.Bind(wx.EVT_MENU, self.Menu2033, id=2033)
117
118 self.Bind(wx.EVT_MENU, self.Menu301To303, id=301)
```

Demo Log Messages

```
11:02:29: OnAppActivate: False
11:02:35: OnActivate: True
11:02:35: OnAppActivate: True
11:02:35: OnActivate: False
11:02:35: OnAppActivate: False
11:02:35: OnActivate: True
11:02:35: OnAppActivate: True
```

Welcome to wxPython 4.0.7.post2

Einbau wxMenu

The screenshot shows the wxPython: (A Demonstration) application window. The title bar reads "wxPython: (A Demonstration)". The menu bar includes "File", "Demo", "Options", and "Help".

On the left is a "wxPython Demos" sidebar with a list of demo items, each with a small icon. The items are: ComboBox, CommandLinkButton, DVC_CustomRenderer, DVC_DataViewModel, DVC_IndexListModel, DVC_ListCtrl, DVC_TreeCtrl, Gauge, Grid, Grid_MegaExample, GridLabelRenderer, ListBox, ListCtrl, ListCtrl_virtual, ListCtrl_edit, **Menu** (highlighted in blue), PopupMenu, PopupWindow, RadioBox, RadioButton, SashWindow, ScrolledWindow, and SearchCtrl. Below the list is a "Filter Demos:" section with a search box containing the text "Suchen".

The main area has three tabs: "Menu Overview", "Demo Code" (selected), and "Demo". Below the tabs are two buttons: "Save Changes" and "Delete Modified".

The "Demo Code" tab displays Python code for menu events. The code is as follows:

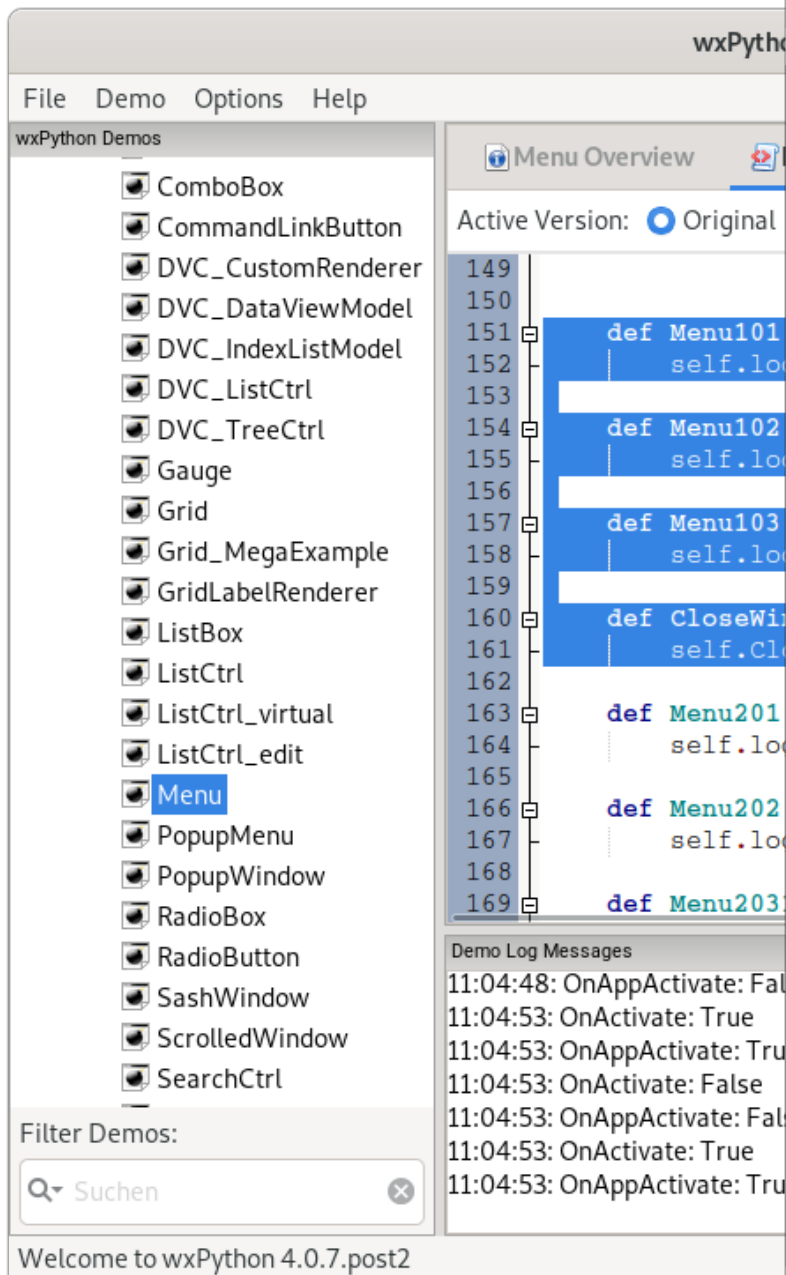
```
149
150
151 | def Menu101(self, event):
152 |     self.log.write('Welcome to Mercury\n')
153
154 | def Menu102(self, event):
155 |     self.log.write('Welcome to Venus\n')
156
157 | def Menu103(self, event):
158 |     self.log.write('Welcome to the Earth\n')
159
160 | def CloseWindow(self, event):
161 |     self.Close()
162
163 | def Menu201(self, event):
164 |     self.log.write('Chemical element number 1\n')
165
166 | def Menu202(self, event):
167 |     self.log.write('Chemical element number 2\n')
168
169 | def Menu2031(self, event):
```

At the bottom of the main area is a "Demo Log Messages" section showing a log of events:

```
11:03:53: OnAppActivate: False
11:04:08: OnActivate: True
11:04:08: OnAppActivate: True
11:04:08: OnActivate: False
11:04:08: OnAppActivate: False
11:04:08: OnActivate: True
11:04:08: OnAppActivate: True
```

The status bar at the bottom of the window displays "Welcome to wxPython 4.0.7.post2".

Einbau wxMenu



```
*Kurs-Version-2.py - /home/nutzer/Dokumente/LI/00/2023-LI-00/Projekte/07/Kurs-Version-2023/Kurs-Version-2.py (3.6.15)* x
File Edit Format Run Options Window Help

# Prepare the menu bar
menuBar = wx.MenuBar()

# 1st menu from left
menu1 = wx.Menu()
menu1.Append(101, "&Mercury", "This the text in the Statusbar")
menu1.Append(102, "&Venus", "")
menu1.Append(103, "&Earth", "You may select Earth too")
menu1.AppendSeparator()
menu1.Append(104, "&Close", "Close this frame")
# Add menu to the menu bar
menuBar.Append(menu1, "&Planets")

self.SetMenuBar(menuBar)

# Menu events
self.Bind(wx.EVT_MENU_HIGHLIGHT_ALL, self.OnMenuHighlight)

self.Bind(wx.EVT_MENU, self.Menu101, id=101)
self.Bind(wx.EVT_MENU, self.Menu102, id=102)
self.Bind(wx.EVT_MENU, self.Menu103, id=103)
self.Bind(wx.EVT_MENU, self.CloseWindow, id=104)

def Menu101(self, event):
    self.log.write('Welcome to Mercury\n')

def Menu102(self, event):
    self.log.write('Welcome to Venus\n')

def Menu103(self, event):
    self.log.write('Welcome to the Earth\n')

def CloseWindow(self, event):
    self.Close()

def OnCloseMe(self, event):
    self.Close(True)

def OnCloseWindow(self, event):
```

Einbau wxMenu

