Assignment 15

GUI

LILLEBAELT ACADEMY OF
PROFESSIONAL HIGHER EDUCATION

Author
Martin Grønholdt
mart80c7@edu.eal.dk

Sunday 26 February 2017

Table of Contents

Introduction

The programs in this hand-in is an example of using classes in Python, the getter/setter patterns is implemented throughout.

All files for this hand in are available at:
https://github.com/deadbok/eal_programming/tree/master/ass15

Error handling

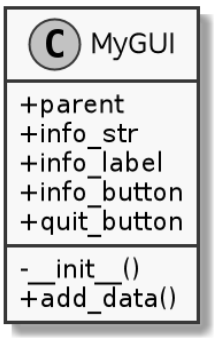
In these particular programs I have focused on the GUI programming part. Error handling is left to the underlying GUI library and Python. This is of course not something that should be done, outside these examples.

1. Name and Address

This program uses a GUI library to create a dialog showing an address when an info button is clicked. There is two version of this program using different GUI libraries. One uses the older Tk toolkit and one uses the never QT5 toolkit. QT5 scales from embedded devices to the desktop, therefore QT5's scope is much larger than Tk, which makes it power full and sometimes fearsome at the same time.

Both programs uses OOP and both generate a label and two button widgets, and lay them out in a window using a grid layout. In both cases an event handler, that fires when the info button is clicked, inserts the address into the label widget.

prog1_tk.py



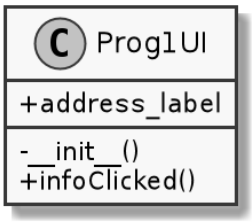
UML graphic representation diagram of the Tk version

Source:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
"""
Name: Program  5 "RetailItem Class"
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (2017-02-19)

Program for entering details in to a ProductionWorker class.
"""
import tkinter
from tkinter import StringVar


class MyGUI:
    def __init__(self, parent):
        """
        Constructor.
        Creates the UI components and layout.

        :param parent: The parant widget.
        """
        # Create the main window widget.
        self.parent = parent
```

```
        # Create a StringVar used to modify the label text.
        self.info_str = StringVar()

        # Create a Label widget
        self.info_label = tkinter.Label(self.parent,
                                        textvariable=self.info_str,
                                        height=3)
        # Place the label at the top of the grid layout
        self.info_label.grid(row=0, column=0, columnspan=3, padx=2, pady=2)

        # Create the buttons.
        self.info_button = tkinter.Button(self.parent,
                                          text='Show info',
                                          command=self.add_data)
        self.quit_button = tkinter.Button(self.parent,
                                          text='Quit',
                                          command=self.parent.destroy)

        # Add the buttons to the second row of the layout grid.
        self.info_button.grid(row=1, column=0, padx=5, pady=5)
        self.quit_button.grid(row=1, column=2, padx=5, pady=5)

    def add_data(self):
        """
        Event handler to add data to the label when the info button is pressed.
        """
        self.info_str.set('Steven Marcus\n' +
                          '274 Baily Drive\n' +
                          'Waynesville, NC 27999')


def main():
    """
    Main function, instantiates TK and the GUI class.
    """
    root = tkinter.Tk()
    # Create an instance of the MyGUI class.
    my_gui = MyGUI(root)
    # Enter the tkinter main loop
    tkinter.mainloop()


# Run this when invoked directly
if __name__ == '__main__':
    main()
```

prog1_qt5.py



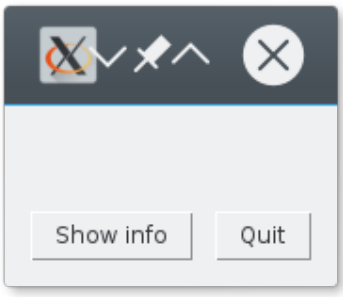
UML graphic representation diagram of the QT5 version

Source:

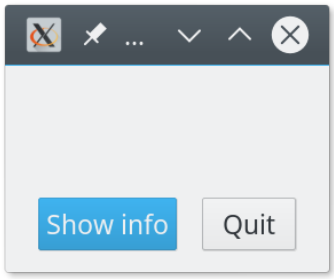
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
"""
Name: Program  5 "RetailItem Class"
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (2017-02-19)

Program for entering details in to a ProductionWorker class.
"""

import sys

from PyQt5.QtWidgets import QWidget, QPushButton, QGridLayout, QApplication, \
    QLabel


class Prog1UI(QWidget):
    """
    Class that creates a simple dialog window.
    """

    def __init__(self):
        """
        Constructor, creates the UI.
        """
        # Call the parant constructor.
        super().__init__()

        # Create the label, using a member variable to be able to reference it
        # later.
        self.address_label = QLabel('\n\n')

        # Create the buttons
        info_button = QPushButton("Show info")
        quit_button = QPushButton("Quit")

        # Create a grid layout
        grid = QGridLayout()
        grid.setSpacing(10)

        # Place the address labe in the top row of the grid and make it span 5
        # columns.
        grid.addWidget(self.address_label, 0, 0, 1, 5)
```

```
        # Place the buttons at the second row, with some cells between them for
        # spacing.
        grid.addWidget(info_button, 1, 1)
        grid.addWidget(quit_button, 1, 3)

        # Connect event handler for button clicks
        info_button.clicked.connect(self.infoClicked)
        # Close the window on clicking "Quit"
        quit_button.clicked.connect(self.close)

        # Set the layout of this widget
        self.setLayout(grid)

        # Set title
        self.setWindowTitle('Program 1 QT5 GUI')
        # Show window
        self.show()

    def infoClicked(self):
        """
        Handler that is called when the info button is clicked
        """
        # Set the text of the label
        self.address_label.setText('Steven Marcus\n' +
                                   '274 Baily Drive\n' +
                                   'Waynesville, NC 27999')


def main():
    """
    Main program
    """
    # Instantiate the QT application class
    app = QApplication(sys.argv)
    # Create out window
    ui = Prog1UI()
    # Exit when done.
    sys.exit(app.exec_())


# Run this when invoked directly
if __name__ == '__main__':
    main()
```

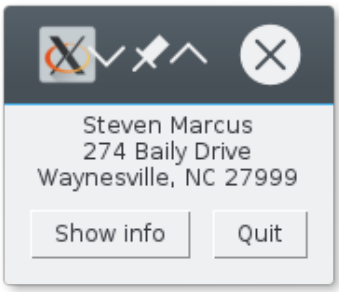
Result



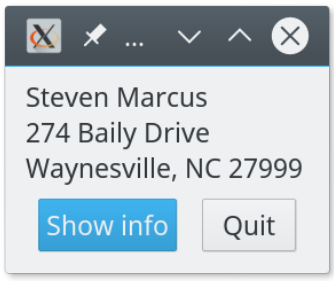
Tk UI before the info button is pressed



QT5 UI before the info button is pressed



Tk UI after the info button is pressed



QT5 UI before the info button is pressed

Notice the title bar, the Tk version does not handle the HiDPI resolution of 3200x1200 that well. It may be possible to tweak this, but QT5 handles this out of the box.

Conclusion

GUI programming in Python is fairly simple and understandable, there are a lot of paradigms that are directly portable from other GUI libraries and languages.