Introduction to QT (and Python GUIs)

Embedded Interface Design

with **Bruce Montgomery**

Learning Objectives

- Students will be able to…
  - Understand what QT and PyQT are and why we use them
  - Create Python based QT interfaces
  - Consider using other Python GUI tools

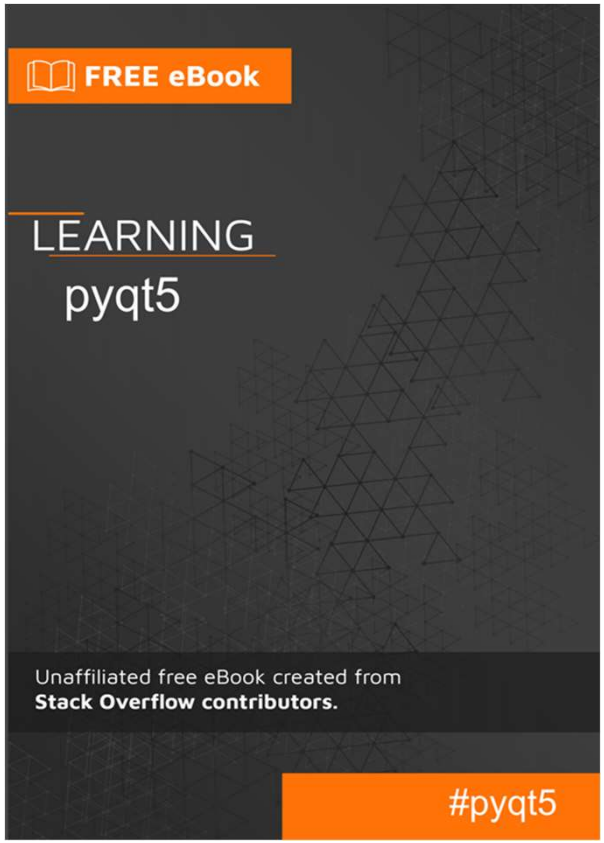
QT?

- QT is a cross-platform library for developing graphical user interfaces for embedded, desktop, and mobile environments
- QT is provided both under commercial and open source licenses (not inexpensive…)
- QT has language bindings for Python, C++, C#, Go, Haskell, JavaScript, Ruby, Rust, and others (under various licenses) [2]
- QT supports GUI development for a variety of operating systems: Windows, Linux/X11, Embedded Linux, QNX, INTEGRITY, iOS, Android [3]

QT World Time Clock Example [1]

About PyQt

- PyQt is a set of open source Python bindings for QT on Windows, OS X, Linux, iOS, and Android [4]
- PyQt5 is the latest and works with QT 5 from the Qt Company
- You can write dialogs in code directly using PyQt or (more often), you can create UIs using the Qt Designer and use PyQt to generate Python code you import to make UIs
- There is a free eBook on PyQt5 here [20]



Quick look at QT Designer

- This will be the QT5 version running on Raspbian (in a VM)…

- Getting PyQT, QT, and the creator/designer tools installed is probably the hardest part of the first project
- Give yourself some time to work through and confirm your installation works!
- Your goal is to have QT5 and PyQT5 both working with Python 3.x

Loading & Using QT5 & PyQT5 on the RPi3

- Install Qt and PyQt
  - sudo apt-get install qt5-default pyqt5-dev pyqt5-dev-tools
  - sudo apt-get install qttools5-dev-tools
- The QT designer executable is found in /usr/lib/i386-linux-gnu/qt5/bin
- Design and save your UI: yourQT.ui
- Compile the ui file to Python:
  - pyuic5 yourQT.ui > yourQT.py
- Import the python file into your main application to access your QT elements:
  - from yourQT import MyDialog
- For a simple example, see [5]

```
import sys
from PyQt5.QtWidgets import QDialog,
Qapplication
from dialog import MyDialog

class AppWindow(QDialog):
    def __init__(self):
        super().__init__()
        self.ui = MyDialog()
        self.ui.setupUi(self)
        self.show()

app = QApplication(sys.argv)
w = AppWindow()
w.show()
sys.exit(app.exec_())
```

Writing Python/QT Code ON the RPi3

- The bundled IDLE3 and Geany Python editors are pretty good
  - You may want to check out other tools like atom.io [6]
  - You can load the Qt5 environment on a laptop and write code there, but be aware, the UI is targeted at the development machine by default – it is tricky to port
- Guidelines for Your First QT Applications
  - General resources and PyQT Tutorials [7]
  - Another example of using QT5 with Python [8]
- If you're working directly on the RPi3, you may want to find an HDMI monitor, a USB or Bluetooth keyboard and a mouse

Writing Python/QT Code ON (or off) the RPi3

- You will likely run a RPi3 Remote Desktop or VNC on your PC or Mac
  - I've done this on my Windows PC using xrdp on the Pi and RDC on the PC, where it worked really well [9]
  - Using QT5 Designer vis remote desktop may require using the noload Welcome option – there is discussion of this on the web [10]
  - TightVNC is recommended for Macs [11]
- You can also run Raspbian in a Virtual Box VM on your laptop – this seems to work well – then transfer files to your RPi3 for final tweaking
  - Loading Raspbian into a Virtual Box VM directions [12]
  - The most current version of the Raspbian x86 desktop is here
  - https://www.raspberrypi.org/downloads/raspberry-pi-desktop/

Other GUI Tools: TKinter

- "TKinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk. Tkinter is not the only GUI Programming toolkit for Python. It is however the most commonly used one." [13]
- TKinter is commonly used for simple native Python GUIs
- An example for the RPi3 [14]
- Tcl/Tk is available as a GUI development tool for many platforms and languages
- We will not be using TKinter in class, but you could add a TKinter UI to a project, possibly for extra credit

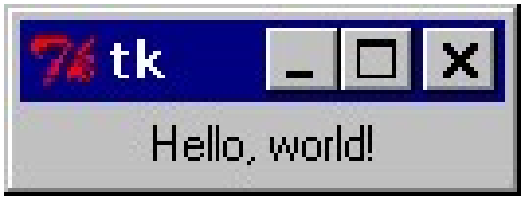
```
from Tkinter import *

root = Tk()

w = Label(root,
text="Hello, world!")
w.pack()

root.mainloop()
```

TKinter Hello World example [15]



Other GUI Tools for Python

- Kivy – Cross Platform GUI Tool [16]
- wxPython [17]
- pyGtk [18]
- Many others! [19]
- Again, we do not review these in class, but you could use them as part of projects for extra credit. See me if interested.

References

[1] http://doc.qt.io/qt-5/qtdesigner-worldtimeclockbuilder-example.html
[2] https://en.wikipedia.org/wiki/List_of_language_bindings_for_Qt_5
[3] http://doc.qt.io/qt-5/supported-platforms.html
[4] https://riverbankcomputing.com/software/pyqt/intro
[5] https://www.codementor.io/deepaksingh04/design-simple-dialog-using-pyqt5-designer-tool-ajskrd09n
[6] https://atom.io/
[7] https://wiki.python.org/moin/PyQt/Tutorials
[8] https://www.baldengineer.com/raspberry-pi-gui-tutorial.html
[9] http://www.raspberrypiblog.com/2012/10/how-to-setup-remote-desktop-from.html
[10] http://stackoverflow.com/questions/37346305/running-qt-creator-via-remote-desktop
[11] https://smittytone.wordpress.com/2016/03/02/mac_remote_desktop_pi/
[12] http://www.aoakley.com/articles/2017-07-04-raspbian-x86-virtualbox.php
[13] https://wiki.python.org/moin/TkInter
[14] https://medium.com/@ronm333/using-tkinter-to-create-a-gui-graphical-user-interface-on-the-raspberry-pi-b382a6bf6497
[15] http://effbot.org/tkinterbook/tkinter-hello-tkinter.htm
[16] https://kivy.org/#home
[17] https://wxpython.org/pages/overview/#hello-world
[18] https://python-gtk-3-tutorial.readthedocs.io/en/latest/
[19] https://wiki.python.org/moin/GuiProgramming
[20] http://www.riptutorial.com/Download/pyqt5.pdf