

Graphical User Interface Programming (Part 1)

Copyright © 2022 by
Robert M. Dondero, Ph.D.
Princeton University

Objectives

- We will cover:
 - Graphical user interface (GUI) programming

Objectives

- More specifically...
- We will cover:
 - “High-level” GUI programming
- We will not cover:
 - “Low-level” GUI programming

3

Objectives

More specifically...

We will cover:

“High-level” GUI programming
Programming with buttons, menus, text fields, ...

We will not cover:

“Low-level” GUI programming
Drawing lines, circles, rectangles, etc.
See COS 126

Motivation

- **Question:** Why study GUI programming?
- **Answers:**
 - Mainstream “advanced programming”
 - Important
 - Illustrates event-driven programming
 - Illustrates concurrent programming

4

Motivation

Question: Why study GUI programming?

Answer 1: Mainstream “advanced programming”

This course is about “advanced programming techniques”
GUI programming clearly falls under that heading

GUI programming is used to create desktop/laptop apps
Granted, most COS 333 projects involve creating web apps, not desktop apps
Nevertheless, there is more to COS 333 than the projects
There is more to advanced programming than composing web apps

Answer 2: Desktop apps are important

Desktop apps are not going away
Desktop editors, ftp clients, terminal apps -- not to mention browsers -- are
here to stay
Somebody must compose them!

Answer 3: Illustrates event-driven programming

Good lead into programming browsers

Answer 4: Illustrates concurrent programming

As we'll see later in the course, GUI programming illustrates multi-threaded programming very well

Agenda

- **Introduction to PyQt5**
- GUI programming in COS 333
- PyQt5 widgets
- PyQt5 layout managers

Intro to PyQt5

- **Qt**
 - Haavard Nord and Eirik Chambe-Eng
 - Trolltech, Nokia, The Qt Company



6

Introduction to PyQt5

Qt

“Cute”

Haavard Nord and Eirik Chambe-Eng

Trolltech, Nokia, The Qt Company

Very successful cross-platform GUI library

Written in C++

440 classes; > 6000 functions/methods

Intro to PyQt5

- **PyQt5**
 - Phil Thompson
 - Riverbank Computing
 - A “Python binding” of Qt

Intro to PyQt5

- **Question:** Why study PyQt5?
 - Instead of some other GUI library?
- **Answers:**
 - Well designed, stable, robust
 - Written in Python

Intro to PyQt5

- **PyQt5** \approx **PySide2**
- In COS 333:
 - **Write** PyQt5 code
 - **Read** PyQt5 or PySide2 docs

9

Introduction to PyQt5

There are two popular Python bindings of Qt:

PyQt5 and PySide2

They're very similar

In COS 333 you should write PyQt5 code

But to learn more, it would be OK to read the documentation for either

A slide near the end of the GUI Pgmning (Part 2) lecture provides references to the documentation

Intro to PyQt5

- See **hellopyqt.py**
 - The job:
 - Display “Hello, world” in a window

Intro to PyQt5

See [hellopyqt.py](#) (cont.)

\$ python hellopyqt5.py

QLabel

11

Introduction to PyQt5

[see slide]

Code notes:

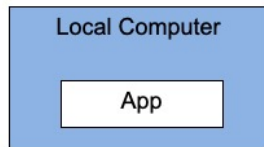
- Instantiate `QApplication` object
 - Must do so before instantiating any widgets
 - Could pass command-line arguments; beyond our scope
- Instantiate `QLabel` object
- Instantiate `QGridLayout` object
 - Add `QLabel` object to it
 - More on layout managers soon
- Instantiate `QFrame` object
 - Add `QGridLayout` object to it
- Instantiate `QMainWindow` object
 - Add `QFrame` object to it
 - Set its initial size
- Make the `QMainWindow` object visible
- `app.exec_()` method call starts event handling loop
 - Whatever value it returns should be the exit status of the program
- Suggestion: accept as a pattern

Agenda

- Introduction to PyQt5
- **GUI programming in COS 333**
- PyQt5 widgets
- PyQt5 layout managers

GUI Pgmming in COS 333

Option 1: Run on your computer



13

GUI Programming in COS 333

Option 1: Run on your computer

[see slide]

Must install Python 3.8

Must install PyQt5

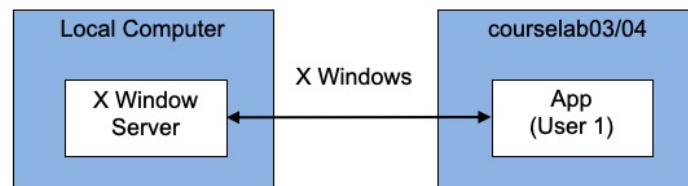
See *A COS 333 Programming Environment* document

Caveat:

Assignments must work on courselab too

GUI Pgmming in COS 333

Option 2: Run on courselab



14

GUI Programming in COS 333

Option 2: Run on courselab

[see slide]

Must install X Window System Server on local computer

See instructions in the *A COS 333 Programming Environment* document from 1st lecture

Your application, while running on courselab, displays its window(s) on your local computer

Agenda

- Introduction to PyQt5
- GUI programming in COS 333
- **PyQt5 widgets**
- PyQt5 layout managers

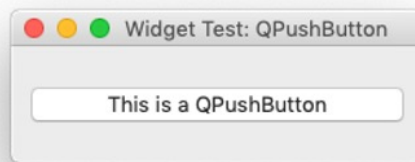
PyQt5 Widget

- See **widgetqlabel.py**
 - The job:
 - Illustrate the **QLabel** widget class



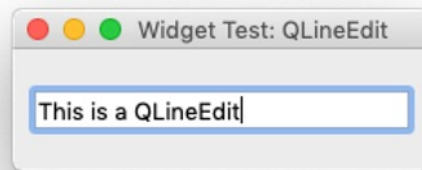
PyQt5 Widgets

- See **widgetqpushbutton.py**
 - The job:
 - Illustrate the **QPushButton** widget class



PyQt5 Widgets

- See **widgetqlineedit.py**
 - The job:
 - Illustrate the **QLineEdit** widget class



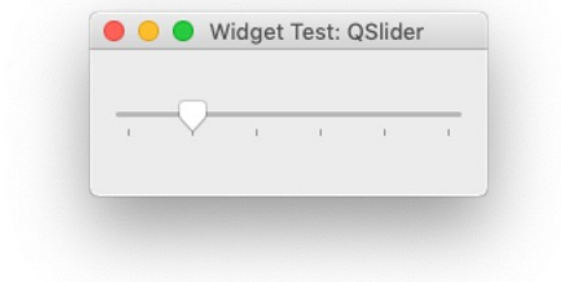
PyQt5 Widgets

- See **widgetqtextedit.py**
 - The job:
 - Illustrate the **QTextEdit** widget class



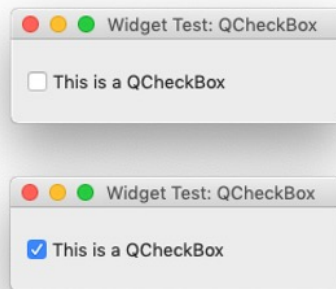
PyQt5 Widgets

- See **widgetqslider.py**
 - The job:
 - Illustrate the **QSlider** widget class



PyQt5 Widgets

- See **widgetcheckbox.py**
 - The job:
 - Illustrate the **QCheckBox** widget class



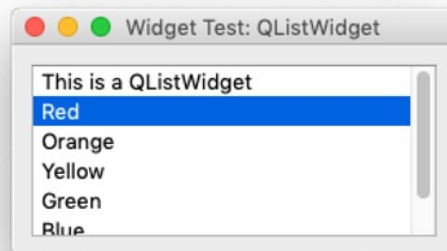
PyQt5 Widgets

- See **widgetqradiobuttons.py**
 - The job:
 - Illustrate the **QRadioButton** widget class



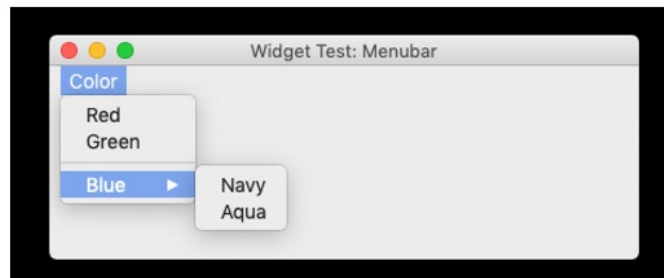
PyQt5 Widgets

- See **widgetqlistwidget.py**
 - The job:
 - Illustrate the **QListWidget** widget class



PyQt5 Widgets

- See **widgetmenubar.py**
 - The job:
 - Illustrate menu bar



PyQt5 Widgets

Conceptual Widget	Relevant PyQt5 Classes
Container	QMainWindow, QFrame
Button	QPushButton
Label	QLabel
Text field	QLineEdit
Text area	QTextEdit
Scrolling text area	QTextEdit
Slider	QSlider
Check box	QCheckBox
Radio button	QRadioButton
List box	QListWidget
Scrolling list box	QListWidget

And many many more

25

PyQt5 Widgets

Any GUI library implements “conceptual widgets” via specific classes

In PyQt5...

[see slide]

Agenda

- Introduction to PyQt5
- GUI programming in COS 333
- PyQt5 widgets
- **PyQt5 layout managers**

26

PyQt5 Layout Managers

Q: How to arrange components/widgets within their containers?

Q: How to specify the resizing behavior of components/widgets?

A: Layout managers

PyQt5 Layout Managers

- See **layoutvbox.py**
 - The job:
 - Display three QLabel objects arranged vertically
 - Illustrate the **QVBoxLayout** layout mgr



27

PyQt5 Layout Managers

[see slide]

Code notes:

Setting the background color of a widget

`QVBoxLayout` class

`addWidget()` method

PyQt5 Layout Managers

- See **layouthbox.py**
 - The job:
 - Display three QLabel objects arranged horizontally
 - Illustrate the **QHBoxLayout** layout mgr



28

PyQt5 Layout Managers

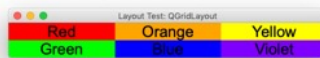
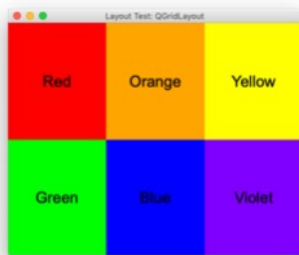
[see slide]

Code notes:

`QHBoxLayout` class

PyQt5 Layout Managers

- See **layoutgrid.py**
 - The job:
 - Display six QLabel objects arranged in a 2x3 grid
 - Illustrate the **QGridLayout** layout mgr



29

PyQt5 Layout Managers

[see slide]

Code notes:

```
QGridLayout class  
addWidget(row,col) method  
setSpacing(i) method
```

PyQt5 Layout Managers

- See [layoutgridstretch.py](#)
 - The job:
 - Illustrate fancy functionality of the **QGridLayout** layout mgr



30

PyQt5 Layout Managers

[see slide]

The job (cont.):

Five QLabel objects: north, south, west, east, center

North and **south**: expand/contract **horizontally**

West and **east**: expand/contract **vertically**

Center: expands/contracts **horizontally** and **vertically**

Code notes:

Can specify row/column spans

`addWidget(row, col, rowSpan, colSpan)` method

Can specify row/column stretches

`setRowStretch(row, stretch)` method

`setColumnStretch(col, stretch)` method

Common layout

PyQt5 Layout Managers

See [layoutgridstretch.py](#) (cont.)

```
layout.setRowStretch(0, 1)  
layout.setRowStretch(1, 1)  
layout.setRowStretch(2, 1)
```

All 3 rows expand/contract
up & down at the same rate

```
layout.setRowStretch(0, 1)  
layout.setRowStretch(1, 2)  
layout.setRowStretch(2, 1)
```

Row 1 expands/contracts
up & down at twice the rate
of rows 0 and 2

```
layout.setRowStretch(0, 0)  
layout.setRowStretch(1, 1)  
layout.setRowStretch(2, 0)
```

Rows 0 and 2 don't
expand/contract up &
down; row 1 does

PyQt5 Layout Managers

See [layoutgridstretch.py](#) (cont.)

```
layout.setColStretch(0, 1)  
layout.setColStretch(1, 1)  
layout.setColStretch(2, 1)
```

All 3 cols expand/contract
side-to-side at the same
rate

```
layout.setColStretch(0, 1)  
layout.setColStretch(1, 2)  
layout.setColStretch(2, 1)
```

Col 1 expands/contracts
side-to-side at twice the
rate of cols 0 and 2

```
layout.setColStretch(0, 0)  
layout.setColStretch(1, 1)  
layout.setColStretch(2, 0)
```

Cols 0 and 2 don't expand
side-to-side; col 1 does

PyQt5 Layout Managers

```
QObject & QLayoutItem
  QLayout
    QBoxLayout
    QFormLayout
    QGridLayout
    QHBoxLayout
    QStackedLayout
    QVBoxLayout
```

33

PyQt5 Layout Managers

There are several additional layout managers

[see slide]

Indentation indicates inheritance

Summary

- We have covered:
 - Introduction to PyQt5
 - GUI programming in COS 333
 - PyQt5 widgets
 - PyQt5 layout managers