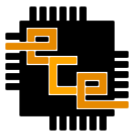
ECE 364
Software Engineering Tools Laboratory

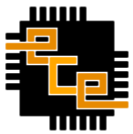
Lecture 10

Python: GUI



Lecture Summary

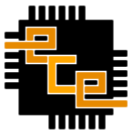
- Introduction to Python GUIs



GUIs in Python

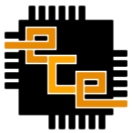
- We use a collection of software libraries that handle the details of drawing elements and implement the low-level functions necessary to run the GUI
  - Usually called toolkits or frameworks

- Most modern GUI toolkits provide the typical elements
  - Window, button, textbox, scrollbar, menu etc.



GUIs in Python (2)

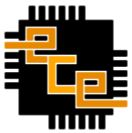
- Common Python GUI toolkits
  - Qt (Python module: `PySide, PyQt`)
  - Gtk+ (Python module: `gtk`)
  - wxWidgets (Python module: `wx`)
  - Tk (Python module: `Tkinter`)
    - The 'standard' Python GUI toolkit



Event Driven Programming

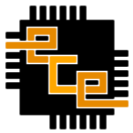
- A program that is controlled by a series of events is called an event driven program

- An event is a thing that happens
  - Mouse clicks on a button
  - Key is pressed on the keyboard
  - A new network packet is received
  - Force is applied to an object (i.e. simulation)
  - A timer “fires” after an interval



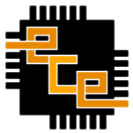
Qt

- Developed by Troll-Tech, bought by Nokia
  - Owned and maintained by Digia in 2010.
  - Became the “Qt Company” in 2014
- Cross-Platform Application Framework
  - Windows, Linux, Mac, Android, iOS …etc.
- Uses C++, but has many language bindings.
- Latest version is Qt 5.x.
- Check http://qt-project.org/ or http://www.qt.io/



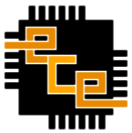
PyQt 5

- Two of the most known Python bindings to Qt are:
  - PySide, maintained by the Qt Project
  - PyQt, maintained by Riverbank Computing
- Both are very similar, but have different licensing options
- Switching between both of them is as easy as changing the import statements



GUI Concepts

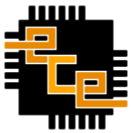
- Widgets: The GUI Elements
- Properties: Attributes of Widgets
- Signals & Slots: Events and Handlers
- Actions: Menu Items, Toolbar Buttons
- Resources: Icons and Images
- Styles: CSS to control the look



Widgets

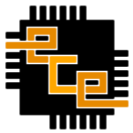
- Widgets are the UI elements of a Qt GUI
  - Each widget is represented by a Python class
  - Each widget has a class hierarchy
  - A new widget is created by instantiating it’s class

- Most widgets are visible GUI elements
  - Buttons, Text box, Windows etc.
  - You have probably seen most of them in other applications



UI 2 Py

- Unlike Visual Studio, or Eclipse, there exists no IDE that supports both, a GUI Designer, and an Editor for Python.
- The QtDesigner Produces a file with `.ui` extension
- These are XML files that contain the layout data.
  - How to make it available to Python?
- Use `pyuic5` to convert the `.ui` to `.py`
  - `pyuic5 <fileName.ui> -o <fileName.py>`
  - Check the full path in the lab
- From there, use the IDE as you normally would.



General Guidelines

- GUI Programming is an advanced topic that requires lots of practice.
- To gain experience:
  - Qt Docs:
    - https://www.riverbankcomputing.com/static/Docs/PyQt4/classes.html
    - https://www.riverbankcomputing.com/static/Docs/PyQt5/api/qtwidgets/qtwidgets-module.html
    - https://doc.qt.io/qtforpython/PySide2/QtWidgets/index.html
    - https://doc.qt.io/qt-5/classes.html
  - Study class hierarchy, mainly under `PyQt5.QtWidgets`
  - Experiment with properties and signals, specially the common ones, like `text` and `clicked`
  - Before you code … Design! (Required in the Project)
  - Start simple, then grow the complexity
  - Worry about style last (We will not require styles in this lab!)
- The second phase of the project is completely about the GUI