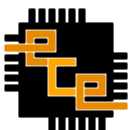




ECE 364

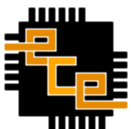
Software Engineering Tools Laboratory

Lecture 10
Python: GUI



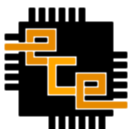
Lecture Summary

- Introduction to Python GUIs



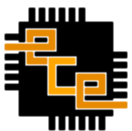
External Resources

- GUIs are very complicated and have many facets of configuration and layout
- **You WILL need to refer to other documentation**, the lecture notes only contain a high-level view of GUI programming with PySide.
- The course staff highly recommend
 - <http://qt-project.org/wiki/PySide>
 - http://qt-project.org/wiki/PySide_Tutorials
 - Rapid GUI Programming with Python and Qt, *by Mark Summerfield*



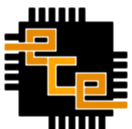
GUIs in Python

- 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.



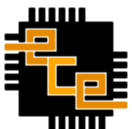
GUIs in Python (2)

- 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



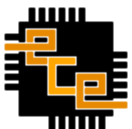
Event Driven Programming

- 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



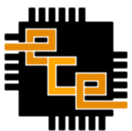
Qt

- Developed by Troll-Tech, bought by Nokia
 - Owned and maintained by Digia since 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, but PySide still uses Qt 4.8.
- Check <http://qt-project.org/> or <http://www.qt.io/>



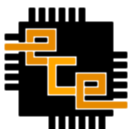
PySide

- 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



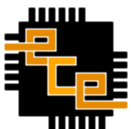
PySide Concepts

- 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

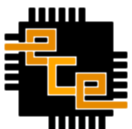


Code-less DEMO: QtDesigner



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 `pyside-uic` to convert the `.ui` to `.py`
 - In the lab, we created the alias: `pysideUIC`
- Use your favorite Python editor from there



General Guidelines

- GUI Programming is an advanced topic that requires lots of practice.
- To gain experience:
 - Study class hierarchy, mainly under `QtCore` and `QtGui`
 - Experiment with properties and signals, specially the common ones, like `text` and `clicked`
 - Start simple, then grow the complexity
 - Worry about style last (but not too late!)
 - Before you code ... Design!
- Your project needs a GUI

