Graphical User Interfaces

Dr. John Artz

Overview

- Event Driven vs. Structured Programming
  - Events and Event Handlers
  - Signals and Slots
- TwoTimes: A Simple GUI
- A Bit About Project1

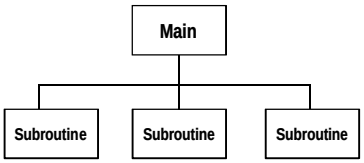
Limited Goals For This Lecture

- Introduce You to Event Driven Programming
- Introduce You to PyQt
- Provide Enough Background So That You Can Provide a GUI Interface for Your Semester Project
- Provide a Basic Background so You Can Pursue Further GUI Development Skills on Your Own if You Choose to do So.

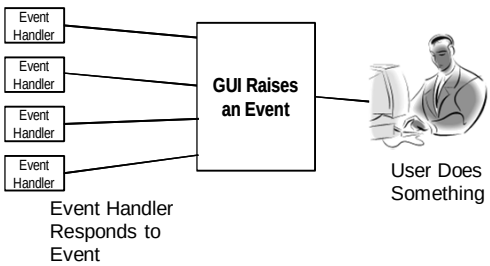
Structured vs. GUI Programming

- In a Structured Program, the Execution of the Program is Controlled by the Programmers
- In GUI Programming, aka Event Driven Programming, the Execution of the Program is Controlled by the User

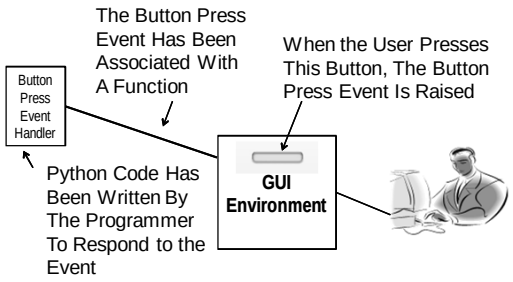
Structured Program



Event Driven Program



More Detail



Signals and Slots

- The Terms Events and Event Handlers is Generic, More Common and Easier to Understand
- PyQT Uses the Terms Signals and Slots
- Signals = Events
- Slots Specify Event Handlers

First Command Line

- As a Point of Reference, We Will Create a Simple Function and Execute It From the Command Line

```
def twotimes(x):
    return 2*x
```

- We Can Type it in at the Command Line

```
twotimes(14)
28
```

Functions vs. Classes

- We Have Already Seen Some Differences Between Functions and Methods
- Now We Need to Establish Some Differences Between Functions and Class

Functions

- We Create a Function Using the Def Keyword
- Once Defined, We Can Invoke the Function Using Its Name
- After the Function Executes, It Returns a Value
- When We Assign a Function to a Variable, the Variable Stores the Returned Value

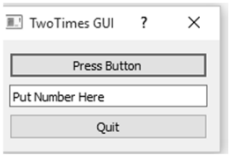
Classes

- A Class is a Template for an Object
- A Class Requires
  - Initialization
  - Internal Variables
  - Methods
- When We Assign a Class to a Variable, that Variable Becomes an Instance of the Class

Now, A Simple GUI

- We Are Going to Begin With a Simple GUI That Pops Up a Window With
  - An Edit Box Where We Can Enter a Number
  - A Button We Can Push to Invoke the twotimes function
- We Will Use It to Learn Some Basic Principles and Expand on it One Step at a Time

The TwoTimes GUI Interface



GUI Program Structure

- Import Needed Libraries
- Define Form Class
  - Form Initializer – Define Widgets and Layout
  - Form Methods – Define Event Handlers
- Application Code
  - Create Application
  - Create and Show Form
  - Initiate Event Handler Loop

Imported Needed Libraries

```
# Import Needed Libraries
import sys
from PyQt4.QtCore import *
from PyQt4.QtGui import *
```

Define the Form as a Class

```
# Define Form as a Class
class Form( QDialog):
    # Form Constructor
    def __init__(self, parent=None):
# Form Methods
    def button1Pressed(self):
    def buttonQuitPressed(self):
# End of Form Class Definition
```

Application Code

```
app = QApplication(sys.argv)    # Create Application
form = Form()                   # Create Instance of Form
form.show()                     # Show the Form
app.exec_()                     # Start Event Handler Loop
```

Form Constructor

```
def __init__(self, parent=None):
    super(Form, self).__init__(parent)
    self.pbutton1 = QPushButton("Press Button")
    self.lineedit1 = QLineEdit("Result Will Appear Here")
    self.pbuttonQuit = QPushButton("Quit")
    layout = QVBoxLayout()
    layout.addWidget(self.pbutton1)
    layout.addWidget(self.lineedit1)
    layout.addWidget(self.pbuttonQuit)
    self.setLayout(layout)
    self.pbutton1.setFocus()
    self.connect(self.pbutton1, SIGNAL("clicked()"),self.button1Pressed)
    self.connect(self.pbuttonQuit, SIGNAL("clicked()"),self.buttonQuitPressed)
    self.setWindowTitle("Simple GUI")
```

Event Handlers

```
# Form Methods
   def button1Pressed(self):
      x1 = int(self.lineedit1.text())
      x2 = twotimes(x1)
      outtext =  str(x1) + " times 2 is " + str(x2)
      self.lineedit1.setText(outtext)
   def buttonQuitPressed(self):
      self.done(1)
      app.quit()
```

TwoTimes

- The Code for Two Times is on Blackboard
- The Code was Written From Scratch for This Example and for the Homework
- You Can Modify It for the Homework
- If You Mess It Up, Go Back and Get a Fresh Copy from Blackboard

Project Interface

Back To ….

The GUI Template Code
Is On Blackboard
You Need to Add Your
Functions to It

Turning in the Project

- Put Everything in a Single Python File Named YourLastName.py (eg. Artz.py)
- Make Sure the TestFile is Read From the Local Directory (i.e. No Pathname)
- Email the File to Me By The Beginning of Our Next Class
- Do Not Zip the File

Some Tips

- Make Sure You Read the Test File from the Local Directory
- Test Each Function to Make Sure it Works
- If You Are Not Developing on a Windows Machine Test Your Code on a Windows Machine to Avoid Any Problems

Helpful Tutorials

- PyQt4 Tutorial http://zetcode.com/gui/pyqt4/
- Let's Learn Python #24 - UI with Python, PyQt & Qt Designer

https://www.youtube.com/watch?v=GLqrzLIIW2E

Helpful Resources

- Rapid GUI Programming with Python and Qt by Mark Summerfield (pdf on Blackboard)
- Introduction to Python Programming and Developing GUI Applications with PyQT by B.M. Harwani