Python
Qt

MACbioIDi – February – March 2018

ULPGC
MAC 2014-2020 Cooperación Territorial
Interreg
Fondo Europeo de Desarrollo Regional
EUROPEAN UNION

Qt Framework

- **Qt** is a cross-platform application development framework:
  - Can be used for application development for a wide range of **target platforms**

- It is a framework written in C++
  - Preprocessor **MOC** (Meta-Object Compiler) is used to extend the C++ language

Qt Framework

- Libraries and development tools in order to develop multiple task:
  - Graphical user interfaces
  - Networking
  - Threads
  - Regular Expressions
  - OpenGL
  - XML...

Qt Framework

- **Qt Company**
  - Started in 1990 by Eirik Chambe-Eng and Haavard Nord as Trolltech
  - Main driver behind Qt
- **Qt Project**
  - A meritocratic, consensus-based community interested in Qt
  - There are many companies and individuals around the globe

Commercial and Open Source

- **Commercial** – costs money
  - Your application can be closed
  - Changes to Qt can be kept closed
- **GPL** – free
  - Your application must be open
  - Changes to Qt must be fedback to the community
- **LGPL** – free
  - Your application can be open or closed
  - Changes to Qt must be fedback to the community

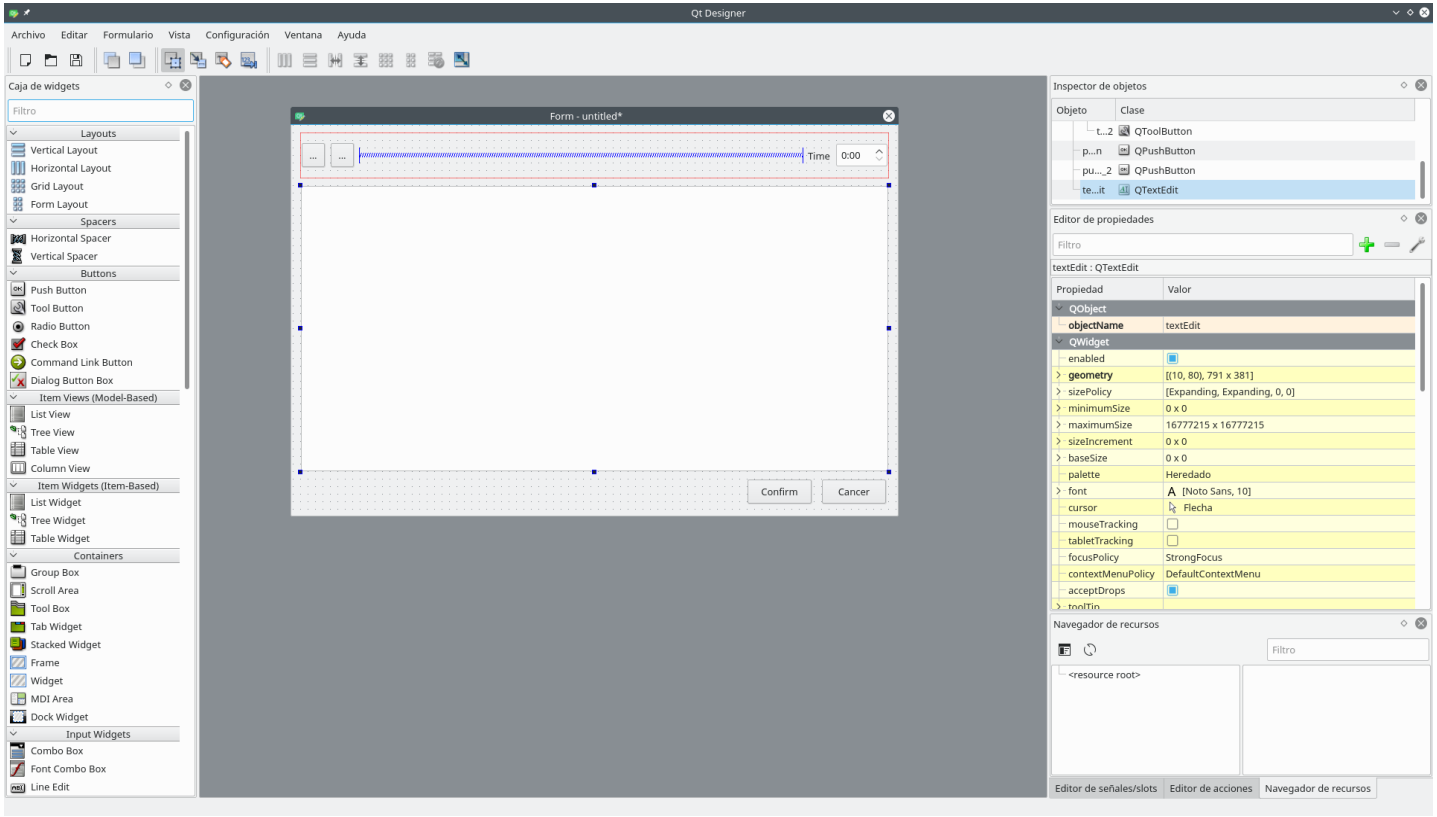
PyQt5

- **PyQt5** is a binding of Qt 5 framework
  - Python Wrapper generated using SIP binding generator

- Components:
  - Modules for different purposes
  - Plugins that enable Qt Designer and qmlscene to be extended using Python code
  - Utility programs:
    - **Pyuic5**
    - **Pyrcc5**
    - **pyupdate5**

Qt Designer

- Qt comes with its own IDE, **Qt Creator**, and a tool to design and building GUIs, **Qt Designer**



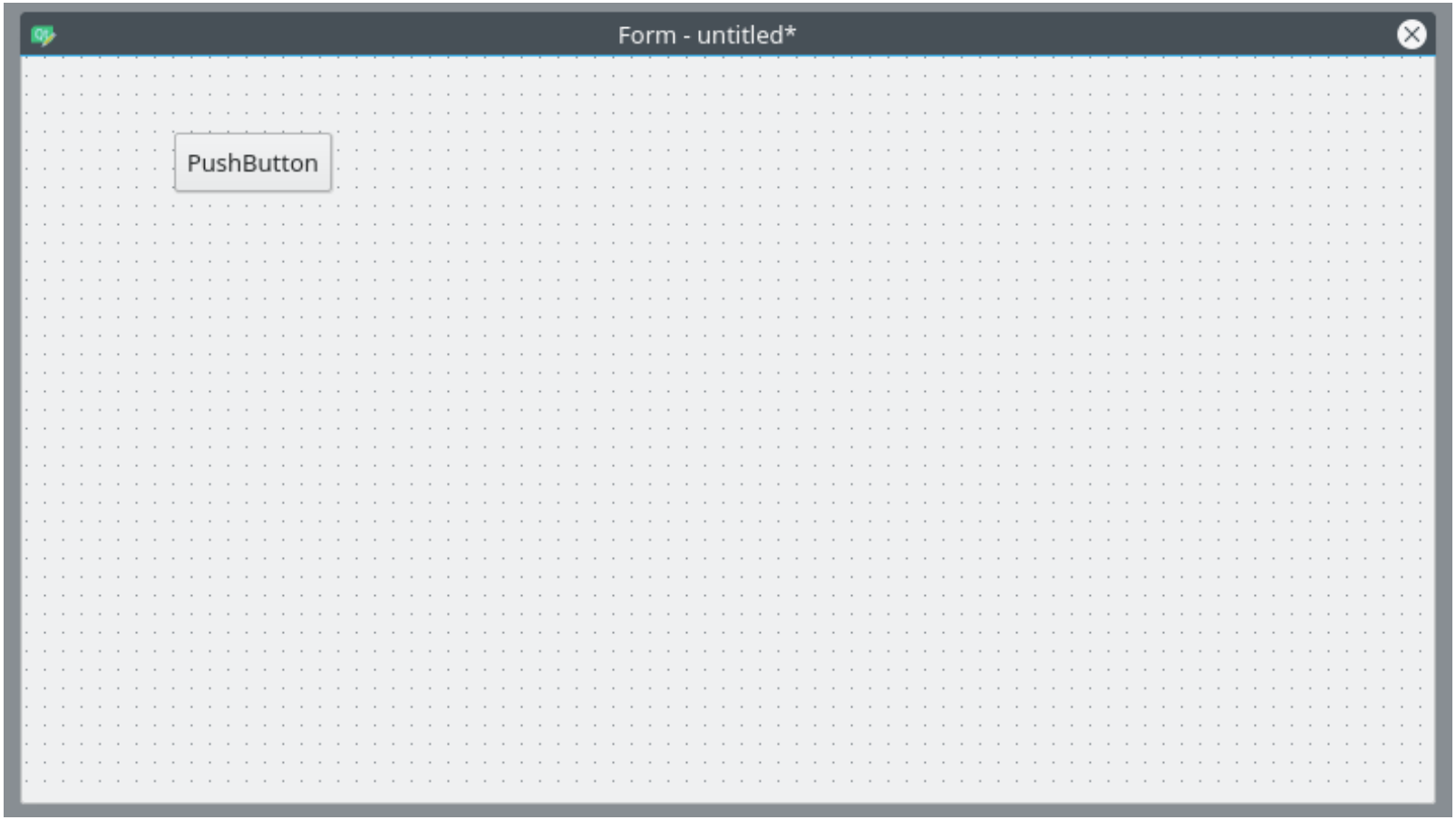
Get familiar with Qt Designer

- Qt Designer generates a XML file
  - Pyuic parse the XML file and generates the python code

- We can generate the python code directly
  - It is a more complicated option to maintain long-term

Example

- We will create a simple widget that contains a Push Button



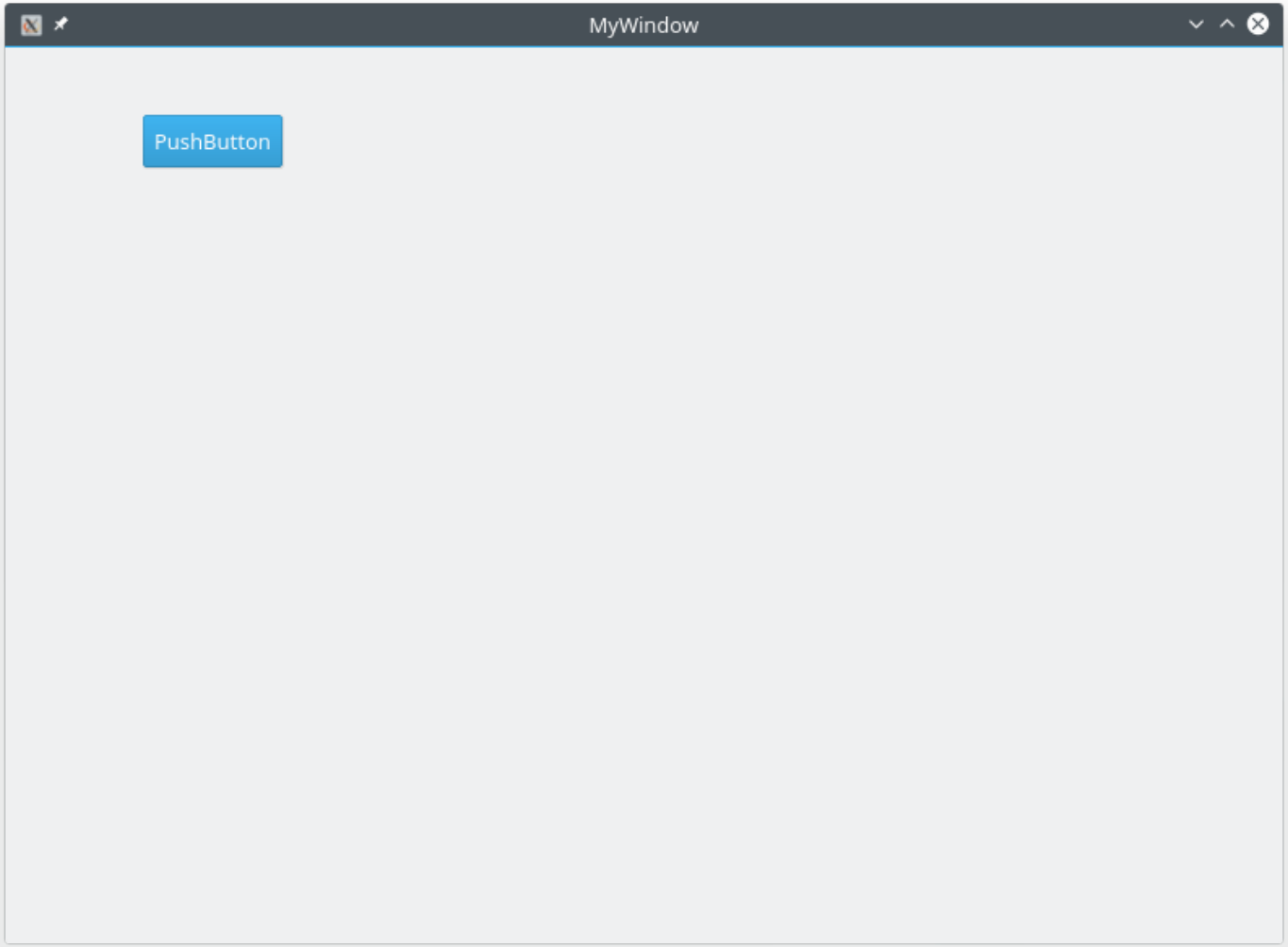
Example

- Create your own window for you application:
  - Create a python class that inherits from QMainWindow
  - Uses the Pyuic5 in order to relation your ui file with the class
    - The ui file was generated with Qt Designer
- Create the QApplication that will show our window

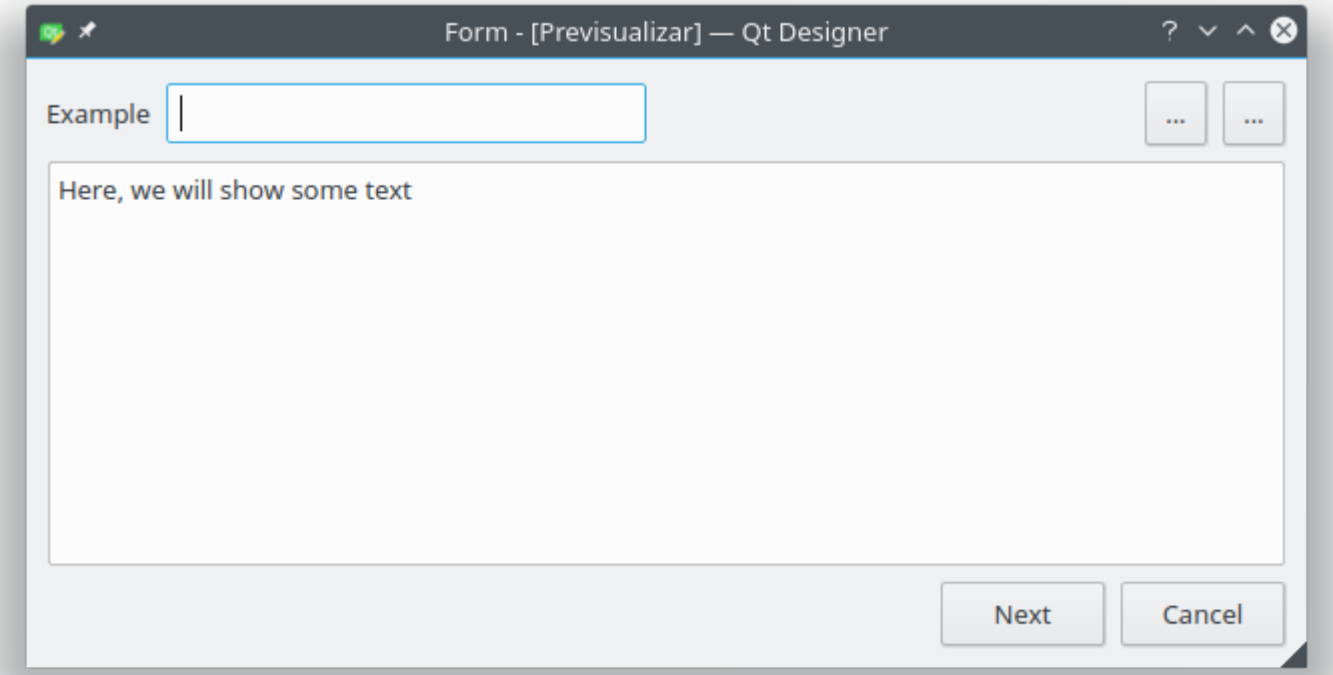
Example

```
import sys
from PyQt5.QtWidgets import QmainWindow
from PyQt5 import uic

class MyWindow(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        uic.loadUi("[Path to your file]", self)
        self.setWindowTitle("MyWindow")

app = QApplication(sys.argv)
mainWindow = MyWindow()
mainWindow.show()
app.exec()
```

Example

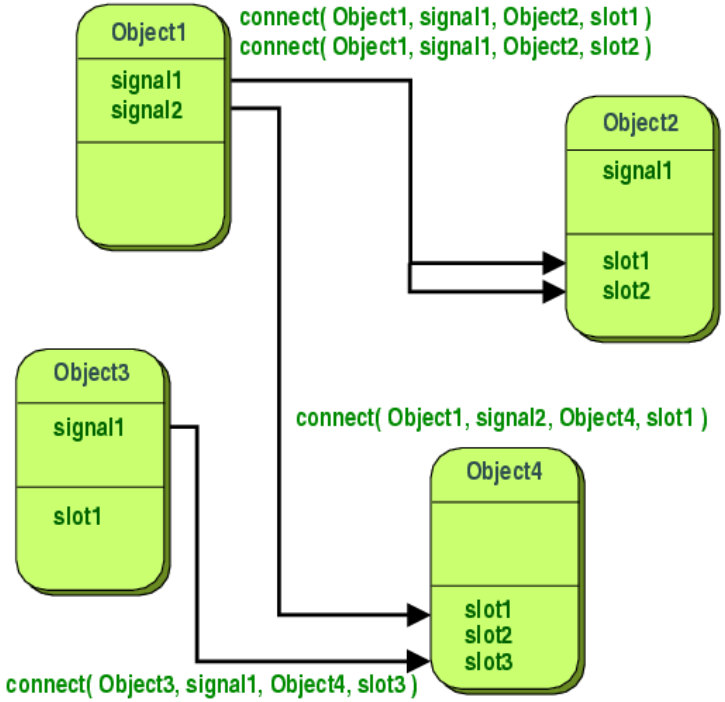


Get familiar with Qt Designer



Signals and Slots

- Used for communication between objects
  - Thanks to Qt's meta-object system (MOC)
  - Signals can transmit variables values



Signals and Slots

- Emitted by an object when its internal state has changed
  - Public access
  - Can be emmited from anywhere
- Signals are automatically generated by the MOC
  - **They can never have return types**
- Slots are called when a signal connected to it is emited
  - Slots can be called like a method too

Signals and Slots Example

```
#include <QObject>

class Counter : public QObject
{
    Q_OBJECT

public:
    Counter() { m_value = 0; }

    int value() const { return m_value; }

public slots:
    void setValue(int value);

signals:
    void valueChanged(int newValue);

private:
    int m_value;
};
```

```
from PyQt5.Qt import QObject, pyqtSlot,
pyqtSignal

class Counter(QObject):
    valueChanged = pyqtSignal(int)

    def __init__(self):
        self.value = 0

    def value(self):
        return self.value

    @pyqtSlot(int)
    def setValue(self, value):
        self.value = value
```

Signal and Slots Exercise

- We will create a **very simple text viewer** from our example generated in this session
  - Assign actions to the QToolButtons and QpushButtons
  - Generates a new thread in order to read the files
    - It will not block the GUI if we read a larger file
- Take a look to:
  - QFileDialog.getOpenFilename()
  - QFile class
  - QAction class
  - QPushButton class
  - QThread class

Python
Qt

MACbioIDi – February – March 2018

ULPGC
MAC 2014-2020 Cooperación Territorial
Interreg
Fondo Europeo de Desarrollo Regional
EUROPEAN UNION