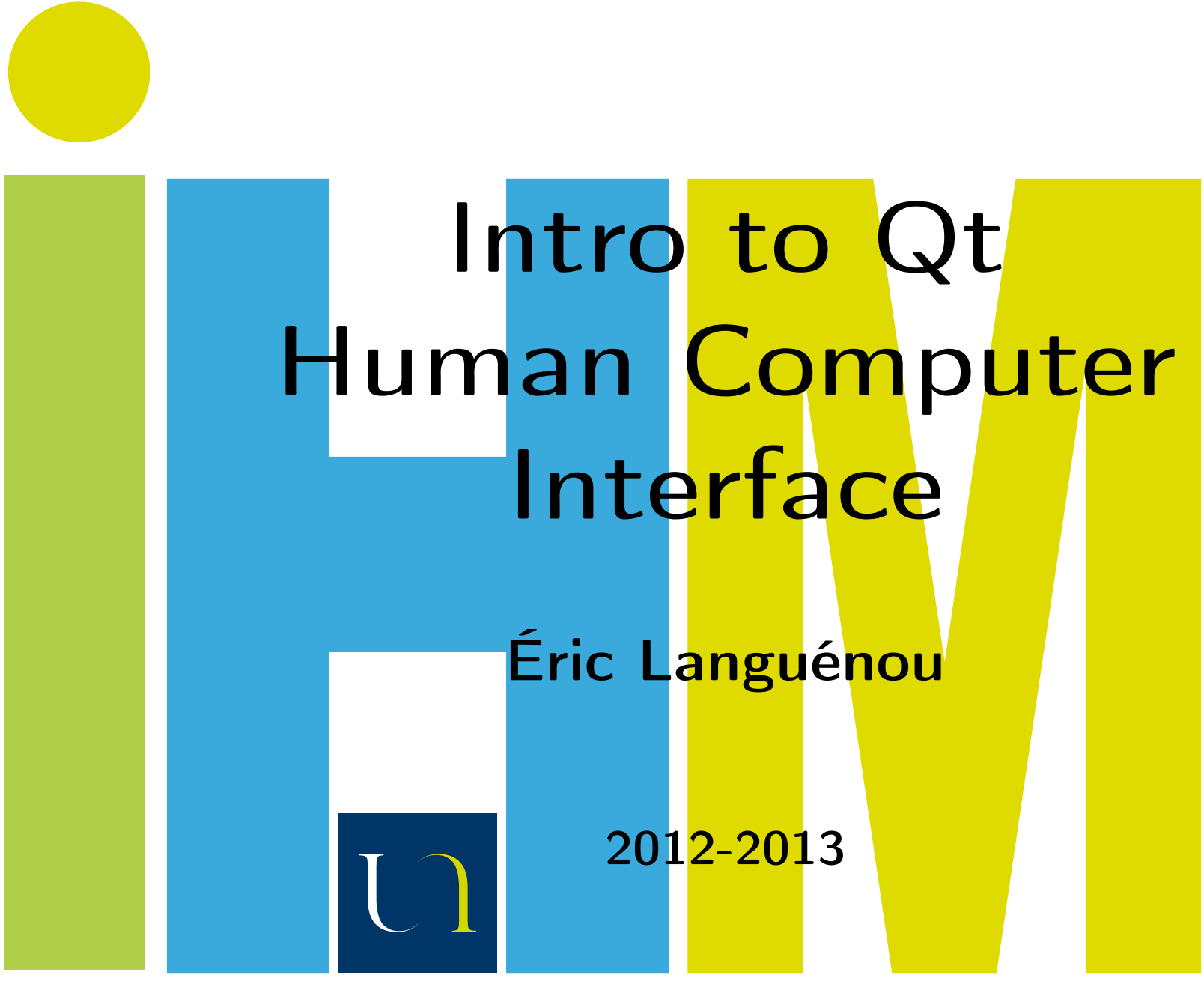


Intro to Qt Human Computer Interface

Éric Languénou

2012-2013

UNIVERSITÉ DE NANTES

What is Qt?

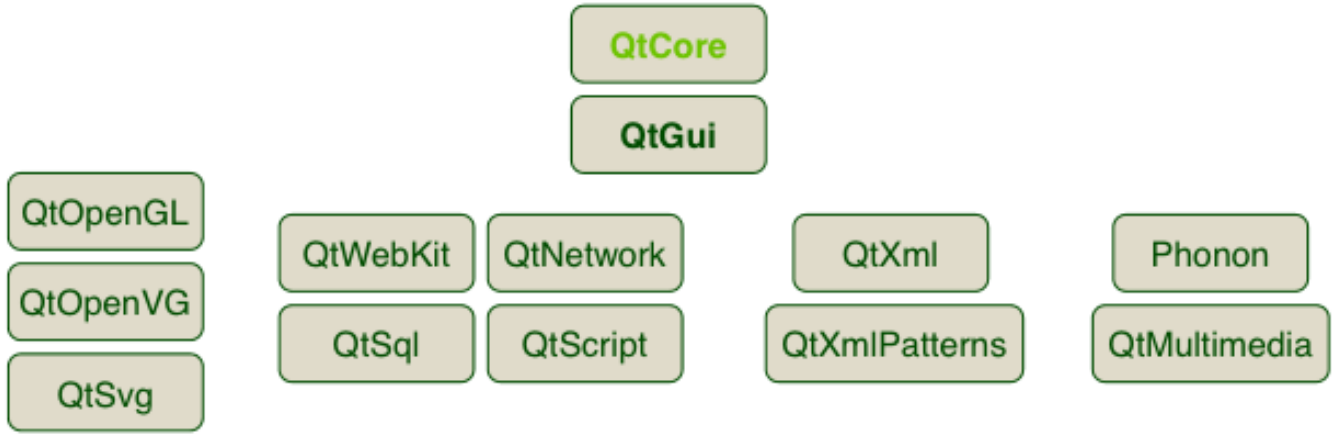
UNIVERSITÉ DE NANTES

Qt a dev framework

- Qt is a cross platform development framework written in C++.
- C++ framework ? bindings for other languages Python, Ruby, C ♯, etc.
- Originally for user interfaces ? now for everything
- Databases, XML, WebKit, multimedia, networking, OpenGL, scripting, non-GUI, etc.

Qt is made up of modules

UNIVERSITÉ DE NANTES



Modules

All modules have a common scheme and are built from the same API design ideas figure modules

Qt extends C++ with macros and introspection

UNIVERSITÉ DE NANTES

```
1 foreach (int value, intList) { ... }
2 QObject *o = new QPushButton;
3 o->metaObject()->className(); // returns "QPushButton"
4 connect(button, SIGNAL(clicked()), window, SLOT(close
      ()));
```

C++

UNIVERSITÉ DE NANTES

- All code is still plain C++
- Cross platform applications built from one source
- Builds native applications with native look and feel
- Easy to (re)use API,high developer productivity, openess, fun to use

Target

UNIVERSITÉ DE NANTES

Desktop target platforms

- Windows
- MacOSX
- Linux/Unix X11

Embedded target platforms

- Windows CE Symbian Maemo
- Embedded Linux Direct framebuffer access

Hello World

UNIVERSITÉ DE NANTES

```
1 #include <QApplication>
2 #include <QLabel>
3 int main( int argc, char **argv )
4 {
5           QApplication app( argc, argv );
6           QLabel l( "Hello World!" );
7           l.show();
8           return app.exec();
9 }
```

Rights : Commercial and Open Source

UNIVERSITÉ DE NANTES

LGPL : free

- Your application can be open or closed
- Changes to Qt must be fed back to the community

GPL : free

- Your application must be open
- Changes to Qt must be fed back to the community

Commercial : costs money

- Your application can be closed
- Changes to Qt can be kept closed

History

UNIVERSITÉ DE NANTES

History

- 1991 : Haavard Nord and Eirik Chambe-Eng, X11 and windows
- 1994 : The company Trolltech
- 1996 : The KDE project
- 2008 : Nokia acquires Trolltech

Qt today

- 840 classes
- 180 developers working on Qt
- Qt community
- KDE built on Qt (KDE community)

Synchronizing Values

UNIVERSITÉ DE NANTES

Connect both ways

```
1   connect(dial1, SIGNAL(valueChanged(int)), dial2,
        SLOT(setValue(int)));
2   connect(dial2, SIGNAL(valueChanged(int)), dial1,
        SLOT(setValue(int)));
```

Infinite loop

- An infinite loop must be stopped.
- no signal is emitted unless an actual change takes place.

```
1 void QDial::setValue(int v) {
2         if(v==m_value) return;
3         ...
4 }
```

Custom signals and slots

UNIVERSITÉ DE NANTES

- A notify signal
- Setters make natural slot
- Signals match the setters
- Q_PROPERTY: special macro that add field with getter, setter, and notifier

```
1  class AngleObject : public QObject {
2          Q_OBJECT
3          Q_PROPERTY(qreal angle READ angle WRITE
               setAngle NOTIFY angleChanged)
4  public:
5          AngleObject(qreal angle, QObject *parent = 0);
6          qreal angle() const;
7  public slots:
8          void setAngle(qreal);
9  signals:
10         void angleChanged(qreal);
11 private:
12         qreal m_angle;
13 };
```

Setter implementation details

UNIVERSITÉ DE NANTES

- Signals are "protected" so you can emit them from derived classes.
- Protection against infinite loops.
- Update the internal state, then emit the signal.

```
1  void AngleObject::setAngle(qreal angle)
2 {
3   if(m_angle == angle)
4     return;
5   m_angle = angle;
6   emit angleChanged(m_angle);
7 }
```

Q_property

UNIVERSITÉ DE NANTES

- Q_PROPERTY(bool enabled READ isEnabled WRITE setEnabled)
- With
  - enabled is boolean field.
  - Accessible by the function isEnabled().
  - Modified by the function setEnabled().

Q_property

UNIVERSITÉ DE NANTES

```
1  class MyClass : public QObject
2  {  Q_OBJECT
3      Q_PROPERTY(Priority priority READ priority WRITE
           setPriority NOTIFY priorityChanged)
4      Q_ENUMS(Priority)
5  public:
6      MyClass(QObject *parent = 0);
7      ~MyClass();
8
9      enum Priority { High, Low, VeryHigh, VeryLow };
10
11     void setPriority(Priority priority)
12     {  m_priority = priority;
13         emit priorityChanged(priority);
14     }
15     Priority priority() const { return m_priority; }
16 signals:
17     void priorityChanged(Priority);
18 private:
19     Priority m_priority;
20 };
```

The Temperature Converter

UNIVERSITÉ DE NANTES

Properties

- Uses the TempConverter class to convert between Celsius and Fahrenheit
- Emits signals when temperature changes

dialog window

contains the following objects

- A TempConverter instance
- Two QGroupBox widgets, each containing
  - A QDial widget
  - A QLCDNumber widget

The Temperature Converter : code1

UNIVERSITÉ DE NANTES

```
1  class TempConverter : public QObject //QObject as
       parent
2  {
3   Q_OBJECT  //macro first
4   public:
5     TempConverter(int tempCelsius,
6                   QObject *parent = 0); //parent
                      pointer
7     //Read and write methods
8     int tempCelsius() const;
9     int tempFahrenheit() const;
10  public slots:
11    void setTempCelsius(int);
12    void setTempFahrenheit(int);
13  signals: //Emitted on changes of the temperature
14    void tempCelsiusChanged(int);
15    void tempFahrenheitChanged(int);
16  private:
17    //Internal representation in integer Celsius.
18    int m_tempCelsius;
19 };
```

The Temperature Converter : code2

UNIVERSITÉ DE NANTES

The setTempCelsius slot:

```
1  void TempConverter::setTempCelsius(int tempCelsius)
2  {
3    //Test for change to break recursion
4    if(m_tempCelsius == tempCelsius)
5      return;
6    m_tempCelsius = tempCelsius; //Update object state
7    //Emit signal(s) reflecting changes
8    emit tempCelsiusChanged(m_tempCelsius);
9    emit tempFahrenheitChanged(tempFahrenheit());
10 }
```

The Temperature Converter : code3

UNIVERSITÉ DE NANTES

The setTempFahrenheit slot:

```
1 //Convert and pass on as Celsius is the internal
      representation
2 void TempConverter::setTempFahrenheit(int
      tempFahrenheit)
3 {
4   int tempCelsius = (5.0/9.0)*(tempFahrenheit-32);
5   setTempCelsius(tempCelsius);
6 }
```

The Temperature Converter : connections

UNIVERSITÉ DE NANTES

- Uses the TempConverter class to convert between Celsius and Fahrenheit;
- Emits signals when temperature changes.

connection associated code

UNIVERSITÉ DE NANTES

```
1  connect(celsiusDial, SIGNAL(valueChanged(int)),
2          tempConverter, SLOT(setTempCelsius(int)));
3  connect(celsiusDial, SIGNAL(valueChanged(int)),
4          celsiusLcd, SLOT(display(int)));
5  connect(tempConverter,SIGNAL(tempCelsiusChanged(int)),
6          celsiusDial, SLOT(setValue(int)));
7  connect(fahrenheitDial, SIGNAL(valueChanged(int)),
8          tempConverter, SLOT(setTempFahrenheit(int)));
9  connect(fahrenheitDial, SIGNAL(valueChanged(int)),
10         fahrenheitLcd, SLOT(display(int)));
11 connect(tempConverter,
12         SIGNAL(tempFahrenheitChanged(int)),
13         fahrenheitDial,
14         SLOT(setValue(int)));
```