

Qt Quick GUI Programming with PySide



Gary Lee

PyCon 2012 Taiwan



Qt Quick

Qt Quick is based on Qt(since v4.7)

Qt Quick consists of a rich set of UI elements

UI is described by QML

UI logic can be controlled by JavaScript

Support signal/slot mechanism to connect Qt

Qt can access properties which defined in QML



Qt Meta Language



Declarative Language

JavaScript based

Inline JavaScript for UI elements

Import JavaScript files

UI behaviors controlled by state machine

UI animation

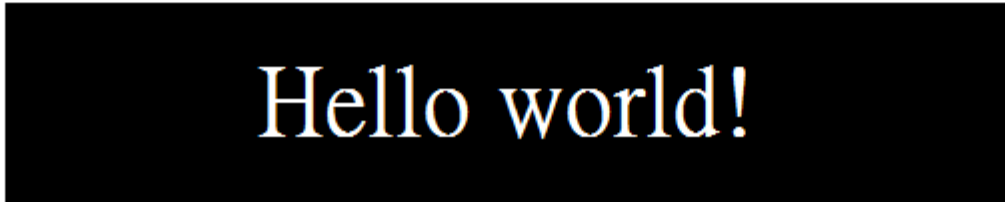


QML at a glance



QML at a glance

```
/* A "Hello world" example for QML */  
  
import QtQuick 1.0  
import "myScript.js" as MyScript  
  
Rectangle {  
    id: helloBox  
    width: 500; height: 100  
    color: "black"  
  
    Text {  
        id: helloText  
        text: "Hello world!"  
        anchors.centerIn: parent  
        color: "white"  
        font.pointSize: 24; font.bold: true  
    }  
}
```



Hello world!





Qt Creator



An integrated development environment (IDE)

Managing Projects

Designing User Interfaces

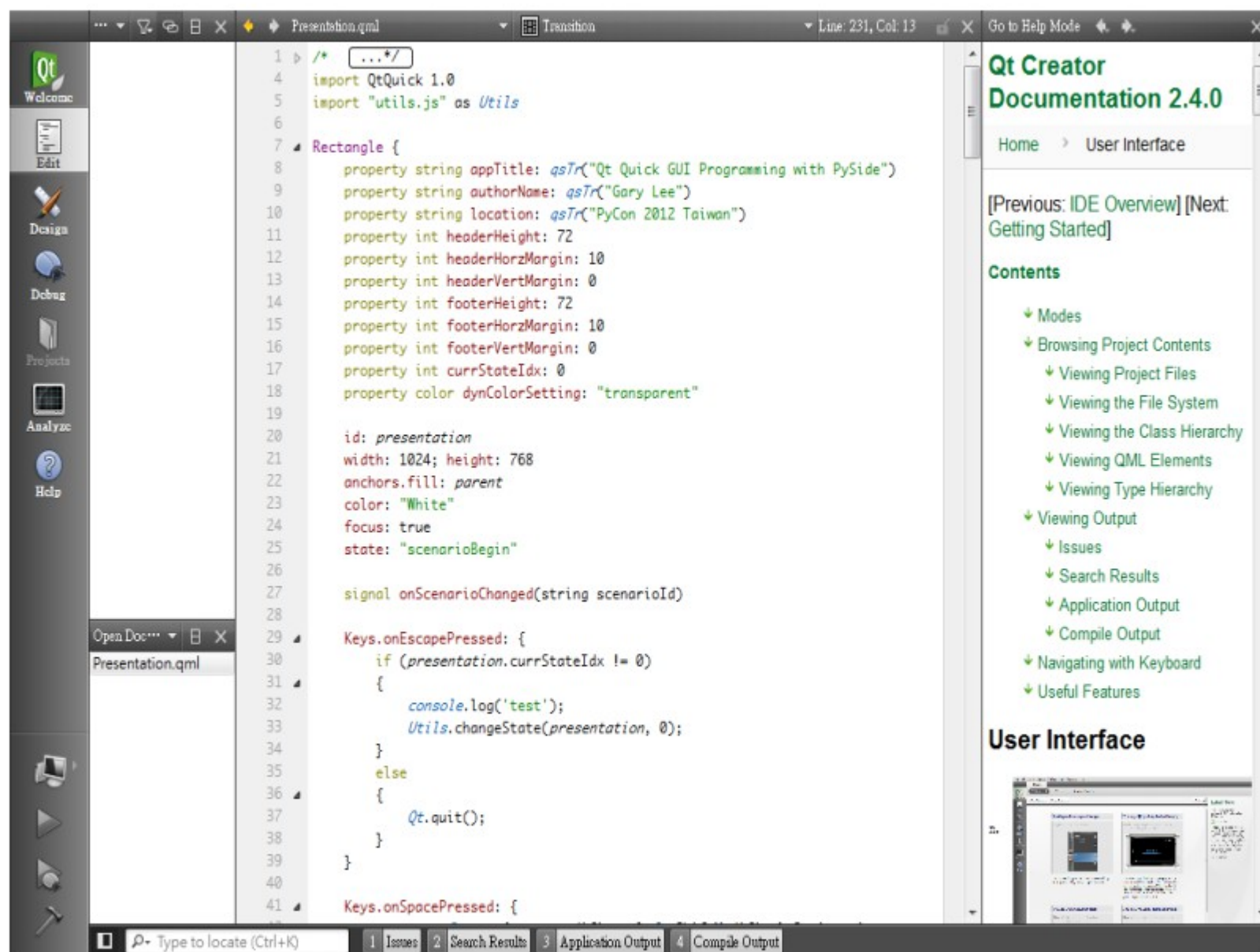
Coding with highlighting, code completion and refactoring actions

Building, Running and Debugging

Desktop and embedded system support

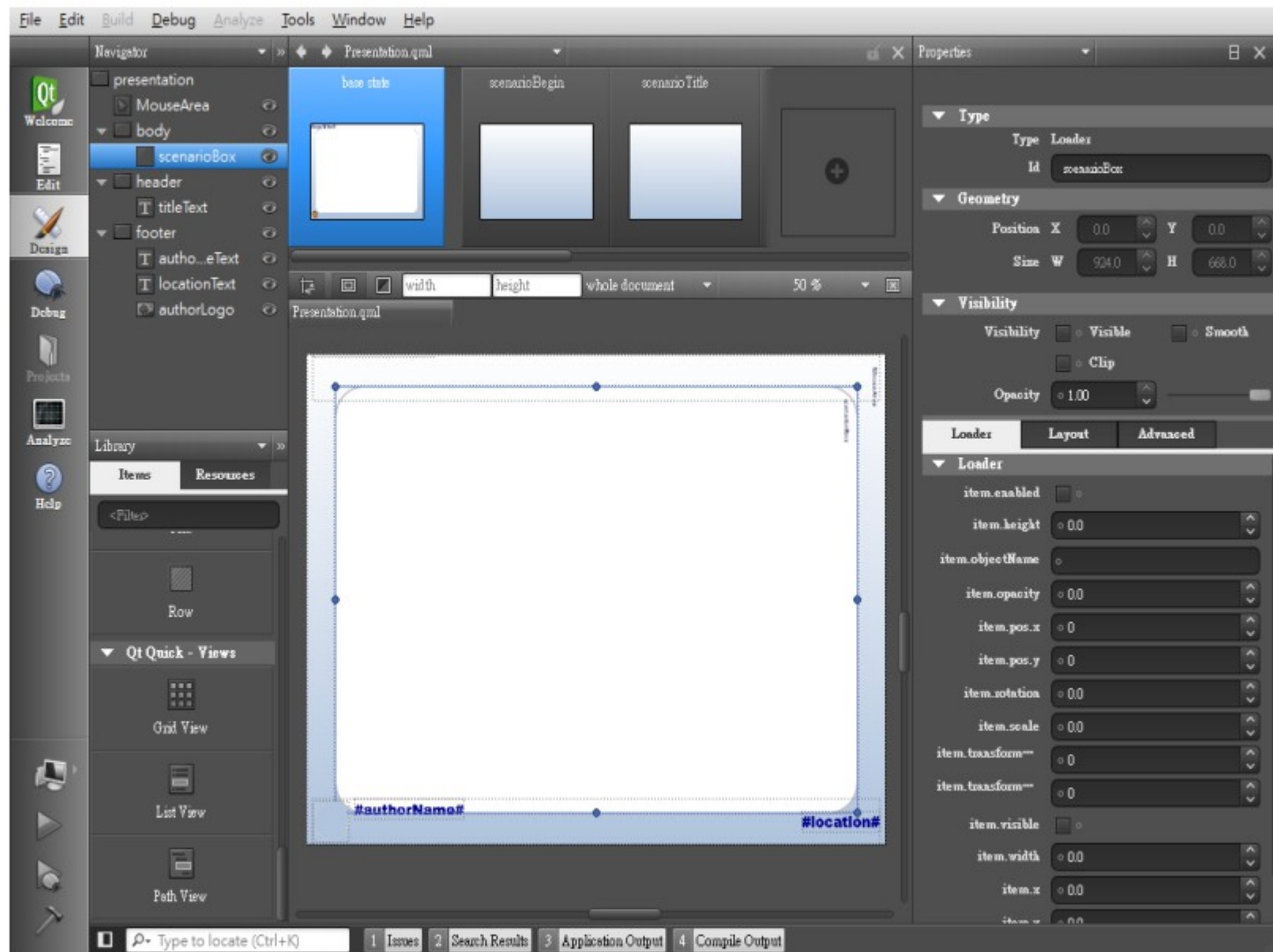


Qt Creator



Gary Lee

Qt Creator



Gary Lee



PySide

A Python binding of Qt toolkit

API compatible with PyQt4

Initiated within the Maemo division of Nokia

Released under the LGPL in August 2009 by Nokia.

**Available for Linux/X11, Maemo 5,
Microsoft Windows, and Mac OS X.**

v1.0 was released on March 04, 2011.

v1.1.0 was released on January 4th, 2012

Became a Qt Add-On on March 6th, 2012.



PySide v.s. PyQt



PySide v.s. PyQt

Import module

```
from PyQt4.QtCore import *  
from PySide.QtCore import *
```

PySide only supports PyQt API 2 (PSEP 101)

No QStrings, QStringList and QVariants

New-style signals and slots

PyQt uses `QtCore.pyqtSignal()` and `QtCore.pyqtSlot()`
PySide uses `QtCore.Signal()` and `QtCore.Slot()` instead

For more information

http://qt-project.org/wiki/Differences_Between_PySide_and_PyQt



PySide
+ **at a glance**
QML



PySide and QML at a glance

```
# "Hello world" example with PySide and QML

import sys
from PySide.QtCore import *
from PySide.QtGui import *
from PySide.QtDeclarative import *

if __name__ == '__main__':

    app = QApplication(sys.argv)
    view = QDeclarativeView()

    view.setSource(QUrl.fromLocalFile('hello.qml'))
    view.setResizeMode(QDeclarativeView.SizeRootObjectToView)

    view.show() # Use view.showFullScreen() for full screen
    app.exec_()
```



Communication between PySide and QML

PySide ↔ **QML**



Communication between PySide and QML

QtCore.Slot and QtCore.Signal can be accessed from QML

QML can defined signal which can be access from PySide

QML functions appear as slots in PySide

context = QDeclarativeView.rootContext()

context.setContextProperty(*propertyName*, *propertyObject*)

http://qt-project.org/wiki/Connecting_QML_Signals_in_PySide

http://qt-project.org/wiki/Using_Qt_Properties_in_PySide



Model View Control



MVC

QML UI

QML Model

Python

Database

AJAX data

**Other
data source**



DEMO

DEMO

4



E N D

