Introduction to Qt

Alexander Færøy

The Camp

July 26, 2010

Outline

What is Qt?

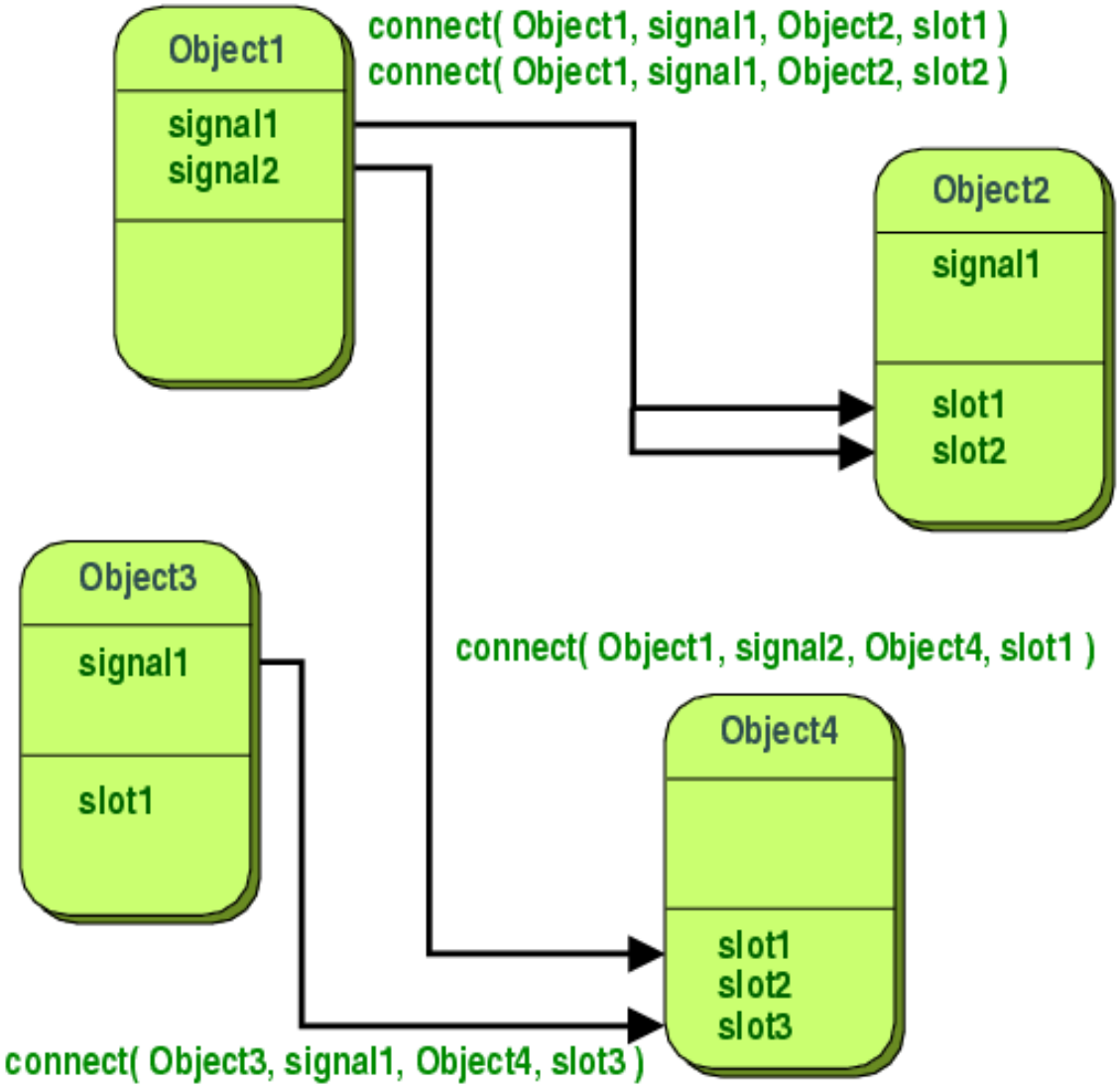
- Cross-platform GUI framework.
- Widget-based.
- C++ framework, but bindings exists for other languages. Alas, not officially supported.
- Used by KDE, Google Earth and Skype.
- "Write once! Run everywhere! After you've recompiled though".

What is Qt?

History

- Project started in 1991 in Norway.
- Dual licensed (LGPL and some fancy commercial license).
- Quasar Technologies, Troll tech, Trolltech and now Nokia.

Signals and Slots



The Qt Object Model

Rationale

- C++ is a very static language; Qt needs to be statically type-checked, but needs a flexible and elegant way of altering objects dynamically.
- Uses smart-pointers over old-school C-style pointers.
- Objects are organized in object trees.
- But why not just use C++'s template-system?

Any Questions?

Conditionals

```
bool a(true);
bool b(false);

if (a) {
    std::cout << "A is true." << std::endl;
} else {
    std::cout << "A is false." << std::endl;
}

if (a && b) {
    std::cout << "Both A and B is true." << std::endl;
} else {
    std::cout << "Either A or B is false." << std::endl;
}

a = false;

if (a) {
    std::cout << "A is true." << std::endl;
} else {
    std::cout << "A is false." << std::endl;
}
```

Variables

```
/* Integer */
int a(10);

/* Boolean data-type */
bool b(true);

/* A String */
QString day("Monday");

/* Floating-point */
float c(13.37);
```

Loops

```
for (int i(0) ; i < 10 ; ++i) {
    std::cout << i << std::endl;
}

bool the_food_is_ready(false);

while (! the_food_is_ready) {
    the_food_is_ready = more_cooking();
}

do {
    std::cout << "This will be executed exactly one time."
        << std::endl;
} while (false);
```

Functions

```
int f(int x) {
    std::cout << "X = " << x << std::endl;

    return x * x;
}

void say(int x) {
    if (x < 10)
        return;

    std::cout << "Number is: " << x << std::endl;
}
```

Namespaces

```
namespace A {
    int a;
}

namespace B {
    int a;
}

a = 1337; // Eeek! Unknown variable!
A::a = 1337;
B::a = 1337;

using namespace A;
a = 1337; // Same as A::a = 1337;

using B::a;
a = 1337; // Same as B::a = 1337;
```

Decent Links

http://www.cplusplus.com/doc/tutorial/
http://doc.trolltech.com/4.6/tutorials.html

Any Questions?