QML for Web

Pietu Pohjalainen

Background

- QT as a cross-platform GUI toolkit has a steep learning curve
  - C++ SDK has the pros and cons of C++
- QML a new(ish) technology for speeding up UI prototyping and application building
- QML is a declarative(ish), JavaScript –compatible language
- Promises reasonable execution speed for good-looking UI's in also in mobile devices
- An introduction available at:

  http://doc.qt.nokia.com/4.7-snapshot/qdeclarativeintroduction.html

The first QML example

```
import Qt 4.7
Rectangle {
  width: 200
  height: 200
  color: "blue"
  Image {
    source:
        "pics/logo.png"
    anchors.centerIn:
        parent
   }
}
```

QT's signals & slots available

- QML builds heavily on using QT's signals and slots,
- E.g. MouseArea

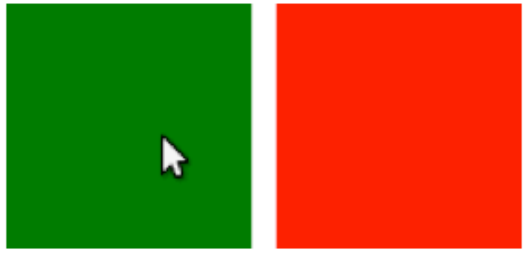
Example Usage

The following example uses a MouseArea in a Rectangle that changes the Rectangle color to red when clicked:

```
import QtQuick 1.0

Rectangle {
    width: 100; height: 100
    color: "green"

    MouseArea {
        anchors.fill: parent
        onClicked: { parent.color = 'red' }
    }
}
```



http://doc.qt.nokia.com/4.7-snapshot/qml-mousearea.html

QMLViewer and example applications

- The QT SDK contains an application QMLViewer for running QML scenes
- A number of QML examples available at:

  http://doc.qt.nokia.com/4.7-snapshot/qdeclarativeexamples.html
- For complex cases, C++ bindings are available
- However, many examples are implemented in pure QML - see e.g.

  http://doc.qt.nokia.com/4.7-snapshot/demos-declarative-calculator.html

Have you spotted the problem?

- A nice programming environment with a lean learning curve
- Builds on JavaScript > familiar to many web devs
- However, it requires specific sdk's installed to your computer
- Promises to run on target machine / cell phone
  - IF QT INSTALLED
- Promises to include web pages, but
  - NOT POSSIBLE TO PLACE A QML APPLICATION TO THE WEB

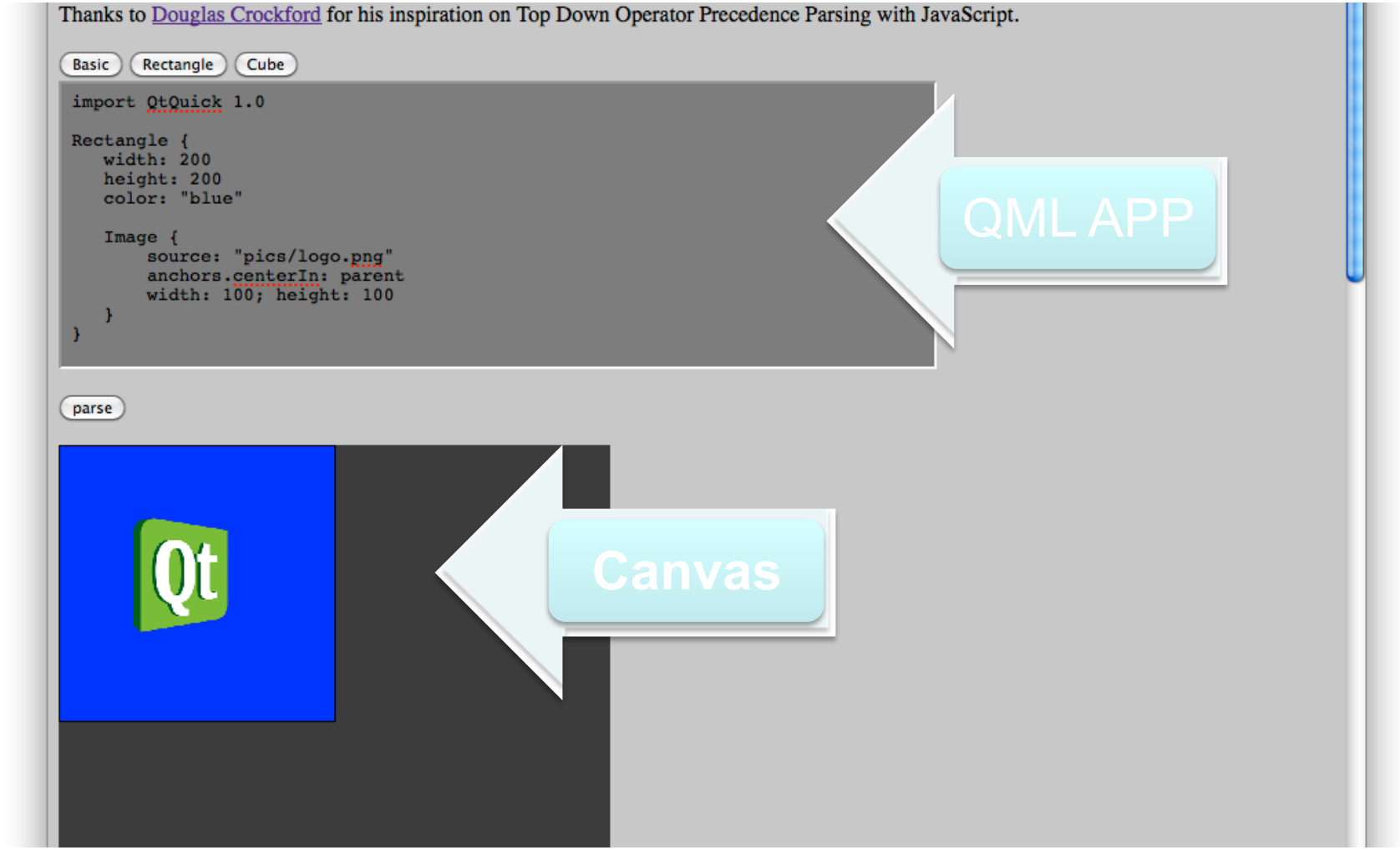
Our contribution: QML 2 WEB

- Our fix: design and implement a QML execution environment that runs in the web browser

1. Parse QML application
2. Construct corresponding JavaScript object tree
3. Render on HTML canvas element

=> PROFIT

QML 2 WEB: An experimental environment for bringing QML apps to the WEB



Why am I presenting this in WebGL forum?

- The implementation employs Processing.JS library for handling the drawing of QML objects
- Actually, our contribution is merely the transformation of QML object tree to JavaScript object tree in format accepted by PJS

1. Processing.JS has (experimental) support for WebGL
2. Qt3D has (experimental) bindings for QML

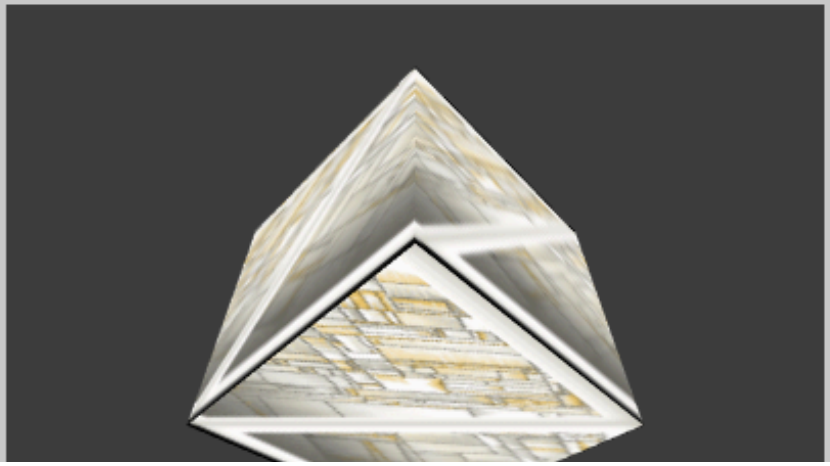
=> OBVIOUS MATCH

Qt3D/QML to WebGL

```
import Qt 4.7
import Qt3D.Shapes 1.0

Cube {
  scale: 90;
  source: "pjs.png";
}
```

parse



Current status of the project

- Currently a hobby project of mine
- Might allow some publishing in CS, e.g.

  Combining Pratt-style parsers; combine a QML parser with the (existing) JavaScript parser
- Only the features shown in the demo work

- Welcome to join the project!

Q&A

- **Discussion**