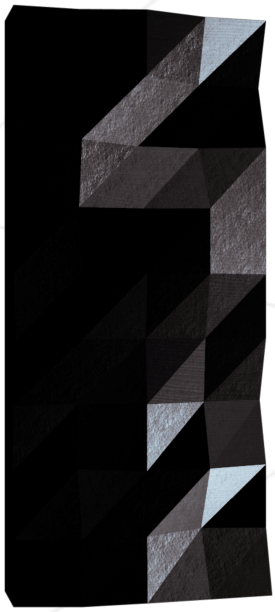

Qt Quick UI

QML essentials

Qt training training@qt.io
42 Pedago Staff pedago@42.fr

Summary: Qt Quick is alternative and completely different way for creating graphical user interfaces in addition to widgets. Both UI techniques are based on forms. Widget forms are defined in XML and converted into C++, if not written in C++ directly. Qt Quick forms are declared with the QML language and interpreted by a QML engine. It is also possible to convert QML code to C++ to avoid run-time interpretation.

Contents

Chapter I

General instructions

Unless explicitely specified, the following rules will apply every day of this Piscine.

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- You must follow the `turn-in process` for each assignment. The url of your `GIT` repository for this day is available on your intranet.
- Your assignments will be evaluated by your Piscine peers.
- In addition to your peers evaluation, a program called the "Moulinette" will also evaluate your assignments. Fully automated, The Moulinette is tough and unforgiving in its evaluations. As a consequence, it is impossible to bargain your grade with it. Uphold the highest level of rigor to avoid unpleasant surprises.
- You must not leave in your turn-in repository any file other than the ones explicitly requested by the assignments.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the `mans` or in the Qt Documentation, which is available both in `QtCreator` IDE and on http://doc.qt.io.
- Remember to use the Piscine forum of your intranet and also Slack!
- You must read the examples thoroughly. They can reveal requirements that are not obvious in the assignment's description.
- Use the latest Qt version. Qt version 5.10 or newer is recommended.

- Many Qt classes are available as standard C++ classes. Qt classes should be preferred to standard classes, as you are supposed to learn Qt.
- Basic Qt coding style should be used, so read `http://wiki.qt.io/Qt_Coding_Style` before writing any assignments.
- Deprecated macros, functions or classes must not be used.
- Any build system, such as `qmake`, `cmake` or Qt Build System, can be used in the assignments. Instructions are based on `qmake`.
- Any editor or IDE, supporting Qt, can be used. However, `QtCreator` usage is strongly recommended.
- By Thor, by Odin! Use your brain!!!

Chapter II

Day-specific instructions

Qt Quick applications are very different from widget applications. This week all assignments are Qt Quick applications, so use the corresponding template in `QtCreator`. In any assignment, you may use `Qt Quick Designer` to create UI forms, but it is possible to write the QML code by hand. In either case, follow the QML coding convention and style.

Chapter III

Assignment 00: Hello world in QML

Implement yet another hello world application with QML. Use a `Text` QML type to show "Hello world!" on the window. The text must be marked localisable. Position the text in the window center.

Chapter IV

Assignment 01: Basic items

Add a gradient background colour to the window. Add a Qt logo from the `res` folder to the application as shown in the figure below.

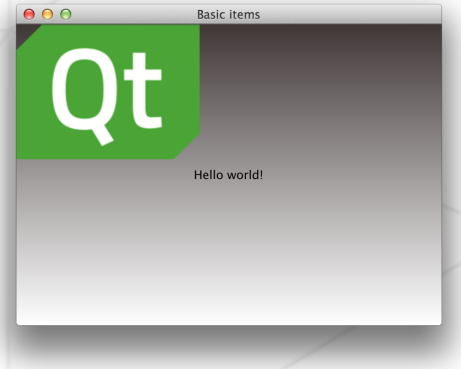

Figure IV.1: Basic items

Chapter V

Assignment 02: Property binding

Continue adding features to the previous assignment. Add a `TextInput` item to your window. The `x` and `y` properties should be 10 per cent of the parent `width` and `height` properties, respectively. The `width` property should be three quarters of the parent width. Use Helvetica font, where the font pixel size is 40 per cent of the parent width, if the parent height is larger than parent width. Otherwise the pixels size is 40 per cent of the parent height. Give the item keyboard focus. If the text is changed, change the text colour to green. Set also the text wrap mode to word wrap.

Pay attention to coding convention.

Chapter VI

Assignment 03: Signal handling

Extend again the previous assignment. Load the Qt logo asynchronously. At start write text "Loading" in the `Text` item. After the image has been loaded, change the text to "Ready". Note that asynchronous image loading may finish before the component gets ready and visible. There is no no need to force the "Loading" text to be visible, if the image loading has already been finished.

Chapter VII

Assignment 04: Focus handling

Create a new QML UI, consisting of a grid of four rectangles as shown below. If a rectangle has the focus, it should be red, otherwise green. The focus can be changed using mouse clicks, any arrow keys or tab and backtab keys. Initially, the focus should be in the top left rectangle.

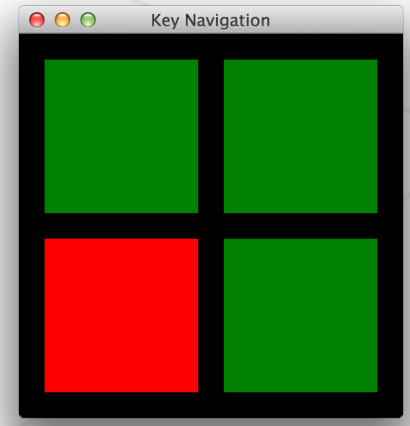

Figure VII.1: Focus handling

Chapter VIII

Assignment 05: Touch and gestures

Write a program three Qt logos.

The logo on the top left corner should support pinch gesture. The pinch gesture can be used to rotate and scale the Qt logo. Two other Qt logos will be visible only, when the window is touched. The logos central point should be positioned in the point, where the touch occurred. If the touch is released, the logos should be hidden.

Log into the debug console also all touch points and their coordinates whenever the touch event is updated.

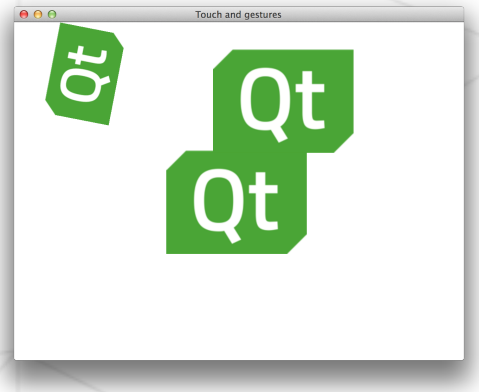

Figure VIII.1: Pinch and multi touch

Chapter IX

Assignment 06: Nested layout

Implement a UI, consisting of nested rectangles as shown below.

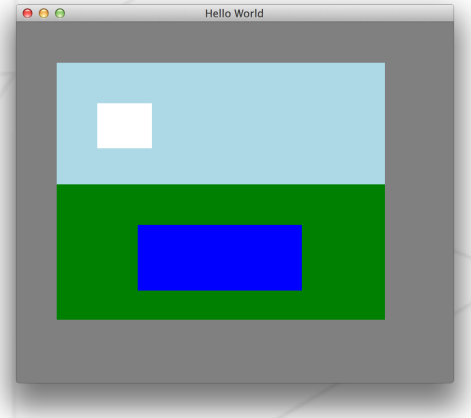

Figure IX.1: Nested layout

The exact geometries of the rectangle are not relevant, but use nested layout to define the positions of all rectangles.

Chapter X

Assignment 07: Scalable UIs

Create a UI with a `Text` item only. Use the `fontSizeMode` property to scale the font pixel size between 5 and 100 pixels, depending on the window size. The font size must not be bound to the window size directly. Use the word wrap as well. The text shown in the window should be "Qt 42".

Chapter XI

Assignment 08: Anchors

Implement a login window as shown below. You can find the images in the `res` folder again. Use anchors to position the login button to the top of the window and the sign out button to the bottom of the window. Margins may have fixed sizes. Anchors also the image and text items inside the login and sign out buttons as well. The buttons should expand horizontally, but have fixed height, as button widgets. The image and text inside the buttons do not need to scale. Usually, this is handled by defining the minimum size for the container item, having buttons. In this assignment, you may simply define the minimum size for the window.

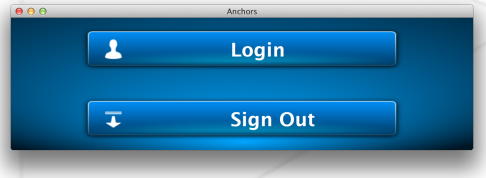

Figure XI.1: Login screen

Chapter XII

Assignment 09: Positioners

Implement similar UI to assignment 06, but this time using positioners. Use `Column` to position light blue and green rectangles. The nested rectangles must be positioned with positioners as well. Any explicit use of `x` or `y` wil result to failure. Anchors can be used, but there must be at least three positioners.