QML Tutorial – version 0.1alpha3

I love lists: to do list, wish list, shopping list etc. For who is always busy, it is a good way to organize important things. That's why I thought that would be fun to make a useful widget to organize a shopping list. Very simple.

Let's start with the basics.

What is a KDE widget?

In the KDE workspaces - *we could say that's your desktop* - are the widgets, the basic "building blocks" of the desktop infrastructure. Widgets power everything, from a weather forecast on your screen to icons, application menus and panels. Adding and removing widgets is extremely easy, so the desktop can be tailored exactly to your needs and desires. You can share your preferred widgets on a local network, access widgets shared by other users and download new widgets with a single click as they become available from the KDE community.

Font: http://www.kde.org/workspaces/

What is QML?

QML is a JavaScript based, declarative language for designing user interface–centric applications. You could compare it to CSS (Cascading Style Sheets). You use CSS to describe the user interface of a webpage, QML describe the user interface of an application, in our case, a KDE widget.

QML elements can be augmented by standard JavaScript both inline and via included .js files. Elements can also be seamlessly integrated and extended by C++ components using the Qt framework. It means, that QML is easy to use, easy to start with, even if you don't know much of programming, but through the links with Javascript and Qt, you can grow your application to do anything.

What is Plasmate?

An application for developing KDE widgets.

You can use a template to create your own widget or you can download the code of a existent widget and then you can quickly your work in the web. Your widget is gonna be available for everyone using KDE.

The cool thing about KDE widgets and QML is...

You can easily develop nice widgets and share it with your friends and with the world!

Logs from #kde-tutorials on Ada Lovelace day

Installing Plasmate:
[19:05] <missingfaktor> sorry to interrupt :) but how do i go about setting up plasmate on my linux mint? or should i wait for the plasmoid tut to start?
[19:06] <camilasan> missingfaktor: good question, let me check
[19:07] <Mte90> i use the version on kde-projects
[19:08] <camilasan> did you try aptget install plasmate?
[19:08] <missingfaktor> camilasan: no i havent. let me try that
[19:08] <missingfaktor> oh. apt-get i did try.
[19:08] <missingfaktor> package not found.
[19:08] <camilasan> or you need to look for a ppa
[19:09] <camilasan> with it
[19:09] <Mte90> for compile does not need too much time
[19:11] <missingfaktor> could not find a ppa with plasmate. i will switch to my other computer that is running suse. it appears plasmate is available on software.opensuse.org.
[19:11] <camilasan> missingfaktor: yes, it is
[19:12] <camilasan> available
[19:12] <jospoortvliet> missingfaktor: you need to find the equivalent of software.opensuse.org for Ubuntu... I dunno what that is...
[19:12]<Mte90> https://projects.kde.org/projects/playground/base/plasmate/repository
[19:12]<jospoortvliet>http://software.opensuse.org/package/plasmate?search_term=plasmate has packages but not for Ubuntu, it seems
[19:13] <jospoortvliet> Mte90: note that that does seem a work in progress, no certainty it will be stable :D
[19:13] <missingfaktor> Mte90: thanks. i have no experience building from source :(
[19:13] <Mte90> i use aptosid and it's not have plasmate on the repo XD
[19:13] <jospoortvliet> Mte90: most distro's have 0.1 alpha2
[19:13] <Mte90> debian sid no
[19:13] <Nightrose> 2 minutes
[19:14] <missingfaktor> jospoortvliet: no problem. i am installing it on suse atm. can continue there.
[19:14] <jospoortvliet> source is here but again, I suggest to try to find packages. openSUSE has them, as missingfaktor just said. Ubuntu might not... http://plasma.kde.org/plasmate/plasmate-0.1alpha2.tar.bz2

[20:26] <jospoortvliet> btw for the openSUSE users - there's a quite stable git snapshot of plasmate which has some nice new features. See on software.opensuse.org/package/plasmate under "Unsupported distributions: > KDE Release 49 openSUSE 12.2".
[20:26] <jospoortvliet> not hugely different, but nice

Starting the tutorial:
[19:14] <camilasan> do you know HTML, CSS or Javascript?
[19:16] <camilasan> If you know HTML, CSS or Javascript, even if just a bit this is gonna be really easy.
[19:16] <camilasan> and do you now what is a kde widget?

An alternative to Plasmate: Plasmoidviewer
[19:16] <Mikey_Chan> Do I need plasmate???
[19:17] <jospoortvliet> it's plasma widget and yes, plasmate makes it a lot easier to play with them
[19:17] <missingfaktor> i dont know css, but know html and js to an extent.
[19:17] <jospoortvliet> Mikey_Chan: depending on your distro there should be a

package somewhere... software.opensuse.org has them for opensuse
[19:17]<camilasan>you can try
http://techbase.kde.org/Development/Tutorials/Plasma/QML/GettingStarted#plasmoidviewer
http://aseigo.blogspot.de/2009/04/plasmoidviewer.html

About KDE Widgets:
[19:18] <camilasan> In the KDE workspaces - we could say that's your desktop - are the widgets
[19:18] <camilasan> from a weather forecast on your screen to icons, application menus and panels
[19:19] <camilasan> in this tutorial I would like to try to help you to build an to do list
[19:19] <camilasan> or whatever-list you like

About QML:
[19:22] <camilasan> QML is a JavaScript based, declarative language for designing user interface–centric applications. You could compare it to CSS
[19:22] <camilasan> but in this case you describe the interface of an application
[19:23] <camilasan> so if you have some knowledge of js css etc you can develop in qml almost intuitivly
[19:24] <camilasan> QML elements can be augmented by standard JavaScript both inline and via included .js files. Elements can also be seamlessly integrated and extended by C++ components using the Qt framework. It means, that QML is easy to use, easy to start with, even if you don't know much of programming, but through the links with Javascript and Qt, you can grow your application to do anything.
[19:24] <camilasan> ok?
[19:24] <missingfaktor_> got it
[19:24] <hcmh> sounds good

Using Plasmate:
[19:25] <camilasan> so lets start opening plasmate
[19:26] <camilasan> http://wstaw.org/m/2012/10/16/CreateaNewAddon.png
[19:26] <camilasan> there is screenshot - first step
[19:27] <camilasan> http://wstaw.org/m/2012/10/16/CreateaNewAddon2.png
[19:27] <camilasan> first step you choose Plasma Widget
[19:27] <camilasan> seconde step you choose the language
[19:27] <camilasan> in our case QML
[19:28] <camilasan> dont forget to give it a name :)
[19:28] <camilasan> http://wstaw.org/m/2012/10/16/CreateaNewAddon3.png
[19:29] <camilasan> now you have a Hello World in QML
[19:29] <camilasan> doubts?
[19:29] <hcmh> no, looks good
[19:29] <missingfaktor_> none so far.
[19:30] <Mte90> ok
[19:30] <hcmh> it also updates the text :-)
[19:30] <camilasan> :)
[19:30] <camilasan> cool
[19:30] <camilasan> http://paste.kde.org/571436
[19:30] <camilasan> so you always will have this main Item
[19:30] <camilasan> (look the link)
[19:31] <camilasan> but the item could be a Rectangle for example

Each component or elements has its properties:
[19:31] <camilasan> so each component has its properties
[19:32] <camilasan> as you can see it in the code, Item, has width and height
[19:32] <camilasan> The Item is the most basic of all visual items in QML
[19:33] <camilasan> let's see a rectangle for example

[19:33] <camilasan> see http://paste.kde.org/571442
[19:34] <camilasan> properties: width, height and color
[19:34] <camilasan> ok?
[19:34] <missingfaktor_> ok
[19:34] <Mte90> ok
[19:34] <hcmh> yes
[19:35] <camilasan> you can just past the rectangle code into the are of the item
[19:35] <camilasan> area*
[19:35] <jospoortvliet> like this: http://wstaw.org/m/2012/10/16/plasma-desktopCO1988.png ???
[19:35] <camilasan> like this http://wstaw.org/m/2012/10/16/plasma-desktopeBa930.png
[19:36] <camilasan> do you see the rectangle
[19:36] <camilasan> ?
[19:36] <camilasan> you should be able to change colors
[19:36] <Mte90> inside the item object
[19:36] <Mte90> okay
[19:36] <missingfaktor_> ok
[19:36] <camilasan> you can give borders
[19:37] <camilasan> border.color: "black"
[19:37] <camilasan> border.width: 5
[19:37] <camilasan> for example
[19:37] <camilasan> so it always work like that
[19:38] <jospoortvliet> yay I got something :D
http://wstaw.org/m/2012/10/16/plasma-desktopGm1988.png

About ID:
[19:38] <camilasan> and for each item you can give it an id
[19:38] <Mte90> it's like jquery
[19:38] <camilasan> an unique id, because later we can manipulate these objects using these ids
[19:38] <camilasan> Mte90: exactly
[19:39] <camilasan> the same concept
[19:40] <camilasan> so let's start with our list
[19:40] <camilasan> or do you want to play more with this rectanble?
[19:40] <missingfaktor_> we can go ahead
[19:42] <camilasan> next step then http://paste.opensuse.org/3975287
[19:43] <camilasan> you can use this code in replacement of the label and the rectangle
[19:43] <camilasan> inside the item
[19:44] <Mte90> ok
[19:44] <missingfaktor_> ok
[19:45] <camilasan> we have a rectangle that contais Text, TextEdit and a Column
[19:46] <camilasan> ok, I am going to paste my complete code so we can go through it and you shouldnt get any errors
[19:47] <camilasan> see the code
[19:47] <camilasan> http://paste.opensuse.org/52715900
[19:47] <jospoortvliet> aah that works
[19:48] <missingfaktor_> working nicely here :) http://i.imgur.com/rVpwr.png
[19:48] <camilasan> nice :)
[19:48] <hcmh> ah, you named the rectangle root and put everything inside... yes, not it works :-)
[19:49] <camilasan> so we have this main Item, wich you can actually do whateber you want, like make it colorful or something
[19:49] <camilasan> and then the rectangle called root
[19:51] <camilasan> and I found more easy to align objects using Column
[19:51] <camilasan> you can take a look at here http://doc.qt.digia.com/4.7-

snapshot/qml-column.html
[19:52] <camilasan> the Column (id column) has the title and an TextEdit wich allows you to input text.
[19:52] <camilasan> ok?
[19:53] <camilasan> Text - you can not edit
[19:53] <camilasan> TextEdit - you can edit
[19:53] <jospoortvliet> you talk about this piece, right: http://wstaw.org/m/2012/10/16/plasma-desktopZz1988.png
[19:53] <camilasan> ok untill now?
[19:53] <hcmh> makes sense, so far
[19:54] <camilasan> jospoortvliet: exactly
[19:54] <camilasan> then below it there is another Rectangle, there is no id for it
[19:55] <camilasan> below the Column

QML Data Models: http://doc.qt.digia.com/4.7-snapshot/qdeclarativemodels.html
[19:55] <missingfaktor_> camilasan: is there a layout manager or something that'll let us specify weight, cell co-ordinates etc for components? something akin to java's gridbaglayout.
[19:55] <camilasan> this rectangle is our button
[19:55] <camilasan> missingfaktor_: yes
[19:56] <camilasan> you can work with gridview for example
[19:56] <camilasan> listview
[19:56] <jospoortvliet> http://doc.qt.digia.com/4.7-snapshot/qdeclarativeelements.html
[19:56] <missingfaktor_> ok
[19:56] <missingfaktor_> :)
[19:57] <camilasan> The Rectangle, our button has its Text "Add Item"
[19:57] <camilasan> then again

Aligment:
[19:57] <camilasan> you can play with the properties: with, height, alignment
[19:59] <camilasan> more about alignment: http://doc.qt.digia.com/qt/qml-text.html#verticalAlignment-prop
[19:59] <camilasan> now about the button

Mousearea:
[20:00] <camilasan> "A MouseArea is an invisible item that is typically used in conjunction with a visible item in order to provide mouse handling for that item. "
[20:00] <camilasan> http://doc.qt.digia.com/4.7-snapshot/qml-mousearea.html
[20:01] <camilasan> in this case MouseArea works with the rectangle
[20:01] <camilasan> ok?
[20:01] <camilasan> the rectangle is the mouse area where you can click
[20:02] <jospoortvliet> so, I'm guessing the 'anchors.fill: parent' is what makes the button be clickable?
[20:02] <jospoortvliet> the rectangle I mean
[20:03] <camilasan> jospoortvliet: no
[20:03] <jospoortvliet> oh...

Anchors:
[20:03] <camilasan> the anchors is a way to control the position of the object, the rectangle
[20:03] <camilasan> "Anchors provide a way to position an item by specifying its relationship with other items."
[20:03] <camilasan> http://doc.qt.digia.com/qt/qml-item.html#anchors.fill-prop
[20:04] <jospoortvliet> ok but then it positions the 'clickable area' to be filling the rectange, right?
[20:04] <camilasan> yes

[20:04] <jospoortvliet> then I understood it, just said it wrong :D
[20:05] <camilasan> in this my example I needed to fight a bit with these anchors
[20:05] <camilasan> ok untill now?
[20:06] <camilasan> playing with the code?
[20:08] <camilasan> again like js, jquery we can handle mouse events

Problems:
[20:08] <hcmh> yes, everything is ok.... but the Qt.createQmlObject looks like trouble... are we gonne change that later on?
[20:08] <camilasan> hcmh: why?
[20:08] <camilasan> there is a option to it
[20:09] <camilasan> http://qt-project.org/doc/qt-4.8/qml-qt.html#createComponent-method
[20:09] <hcmh> because we are creating unnamed objects?
[20:10] <hcmh> which are only going away once we reset the widget..
[20:10] <hcmh> but most likely we are getting there..
[20:11] <camilasan> hcmh: you have a point, that's just the way I manage to make it
[20:12] <camilasan> I am sure there are another 100 ways to make it better
[20:12] <hcmh> ok
[20:12] <camilasan> about this line: Qt.createQmlObject('import QtQuick 1.0;TextEdit {text: "item";color: "black"}', column);
[20:13] <camilasan> you can write whatever qml element you want with this
[20:13] <camilasan> the parameter column is the parent of the TextEdit
[20:15] <camilasan> in this example, the new editable text itens are being included below each other
[20:15] <camilasan> questions?

How to publish your widget?
[20:16] <camilasan> How to publish your widget?
[20:16] <missingfaktor_> no
[20:16] <camilasan> (when you think it is good enough to make it)
[20:17] <camilasan> I mean, you can install it in your own system
[20:17] <camilasan> before you publish it
[20:17] <missingfaktor_> i installed it
[20:18] <camilasan> on the toolbar there is a "Publish" button
[20:18] <missingfaktor_> and added to my desktop.

Errors:
[20:18] <missingfaktor_> the translucent thing in the background doesnt extend to the whole of rectangle :(
[20:18] <camilasan> I believe we can use the anchors to fix it
[20:19] <missingfaktor_> ok
[20:20] <camilasan> so, the idea of the tutorial was of course to get you started
[20:20] <camilasan> we can now figure out how to improve things together - or separate :D
[20:20] <camilasan> I'll stay around for a while...
[20:21] <Mamarok> isn't it more that we didn't define the size of the widget yet?
[20:21] <camilasan> I am trying it too
[20:21] <missingfaktor_> camilasan: thanks so much. :) i learnt about plasmate only today.
[20:22] <camilasan> missingfaktor_: my pleasure. and plasmate makes life easier
[20:23] <camilasan> I hope we can all keep learning from here...
[20:25] <missingfaktor_> yep.

Where do i post my further kde dev questions in general?
[20:25] <missingfaktor_> where do i post my further kde dev questions in general?
[20:26] <camilasan> missingfaktor_: #kde-devel

[20:28] <Nightrose> alright folks :)
[20:28] <Nightrose> thanks so much for participating

Fixing the error:
[20:28] <hcmh> http://imgur.com/gTXm7 it's working :-)
[20:28] <Nightrose> hcmh: wohoooooo
[20:29] <jospoortvliet> hcmh: awesome!
[20:29] <camilasan> hcmh: cute!
[20:29] <camilasan> cool
[20:29] <jospoortvliet> hcmh: how did you do that?
[20:29] <jospoortvliet> could you, like, paste your code?
[20:30] <hcmh> I just made everything transparent and then resized the widget:
[20:30] <hcmh> http://paste.kde.org/571538/
[20:30] <fregl> :)
[20:31] <camilasan> interesting hcmh
[20:31] <jospoortvliet> ha, nice trick
[20:31] <jospoortvliet> but is there now an invisible rectangle still there, or do I not understand this?

How can one make the rectangle resize with the widget?
[20:32] <jospoortvliet> how can one make the rectangle resize with the widget?
[20:32] <fregl> jospoortvliet: "anchors.fill: parent" maybe?
[20:32] <camilasan> I think so too
[20:33] <hcmh> yes, if I make it smaller, add item will be outside..
[20:40] <jospoortvliet> fregl: that works indeed
[20:41] <hcmh> and then we can put " property int minimumWidth: paintedWidth" and "property int minimumHeight: paintedHeight" in the root item and it will always have everything inside of the widget
[20:41] <hcmh> yay :-)
[20:42] <hcmh> scratch that..somehow I can still make it too small..

TextInput x TextEdit:
[20:43] <jospoortvliet> I also just figured out that if you use textInput instead of textEdit it is restircted to one line at a time
[20:43] <jospoortvliet> which is more correct

Links

- http://techbase.kde.org/Development/Tutorials/Plasma/QML/GettingStarted#plasmoidviewer
- http://doc.qt.digia.com/4.7-snapshot/qdeclarativeintroduction.html
- http://kde.org
- http://www.kde.org/workspaces/
- http://techbase.kde.org/Projects/Plasma/PlasMate
- http://doc.qt.digia.com/4.7-snapshot/qdeclarativebasictypes.html
- http://doc.qt.digia.com/4.7-snapshot/qdeclarativeelements.html
- http://doc.qt.digia.com/4.7/qtquick.html
- http://doc.qt.digia.com/4.7-snapshot/qdeclarativeexamples.html
- http://qt-project.org/doc/qt-4.8/qml-anchor-layout.html
- http://techbase.kde.org/Development/Tutorials/Plasma/QML/API