

Marco Martin

Design phase, HIG work in progress

- Based upon Design guidelines of visual Design Group
- https://community.kde.org/KDE_Visual_Design_Group/KirigamiHIG
- The fastest way to have consistent apps with the HIG is to have pre-made controls
- A lot of those controls in kirigami.git
- Plan: make it a tier-1 framework
- Multiplatform: has to work on Desktop Linux, Plasma Mobile, Android, Ubuntu Phone, Windows...

Primitive controls

- What Kirigami is **NOT** about:
 - Bottons, checkboxes etc:
 - Not in topic, use QQC (soon QQC2)
 - Future migration to QQuickControls2 must be as painless as possible

ApplicationWindow

- Base class for applications
- Derives from QQC ApplicationWindow
- Implements some of the central UI elements
- Properties for Drawers
- Passive notifications
- The page navigation looking like a scrollable row (our main way to go back between pages is trough gestures, very effective except corner cases)



Primary Action Button

- Not a class that can be directly instantiated
- Triggers actions and acts as an handle for the side drawers
- The Page provides the actions as a model, Kirigami decides how to visually represent them





- ApplicationWindow supports two drawers that will be overlaid on top of the application contents
- Base class: OverlayDrawer
- It's empty, can be used everywhere
- It can have 4 orientations: left, right, top, bottom
- API compatible with QQuickControls2 Drawer
- Edge slide from left and right, and from bottom on platforms that have sides reserved for the system (Ubuntu, Windows 8-10)

GlobalDrawer

Search (" Search (" Search (" Submenu 1) Submenu 2) Checkable (" Checkable (" Checkab	Widget gall			
Search (iii) Search (iii) Submenu 1) (iii) Submenu 2) (iii) Checkable (iiii) Checkable (iii)	- Waget ga		<	
Search Submenu 1 Submenu 2 Submenu 2 Checkable Checkable Open A Page (iii)			<	
Submenu 1 > Submenu 2 > Image: Checkable (IIII) Image: Open A Page (IIIII) Image: Open A Page (IIIII) Image: Open A Page (IIIII) Image: Open A Page (IIIIIII) Image: Open A Page (IIIIIIII) Image: Open A Page (IIIIIIIIII) Image: Open A Page (IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Search		<	
Submenu 2 (iii) Image: Checkable (iii) Image: Open A Page (iiii) Image: Open A Page (iiiiiiii) Image: Open A Pag	Submenu 1	>	<	
Checkable (iii C Open A Page (iii (iii) (iii)	Submenu 2	>	<	
 Copen A Page (iii) (iii)	□= Checkable		<	
(iii (iii) (ii)) (产 Open A Page		<	
 (iii) 			<	
 (iii) (iii) (iii) Option 1 Option 2 Option 3 (iii) 		_	<	
<pre> Option 1 Option 2 Option 3 </pre>		- 1	(
Option 1 Option 2 Option 3		- 1		
Option 1 Option 2 Option 3		- 8	,	
Option 1 Option 2 Option 3		- 1		
Option 1 Option 2 Option 3		- 1		
Option 1 Option 2 Option 3		- 1		
Option 1 Option 2 Option 3				
Option 2 Option 3	Option 1			
Option 3	Option 2			
	Option 3			:

- Edge slide from left (screenedge or Primary Action button)
- Big pretty title
- Navigable menu, like a menubar
- Bottom and top areas to put arbitrary controls

ContextDrawer

- Edge slide from right (screen edge, Action Button or bottom-right gesture)
- Actions that depend from the current app page: contextdependent
- Bottom-aligned to be thumb-friendly



Custom drawers

Buttor	ns	
	Open Sheet	
	Show Passive Notification	
	Passive Notification Action	
	Disabled Button	
	Tool Button	
	Tool Button non flat	
	\bigcirc	
Button1		
Button2		
	Duttonz	

- The developer can use an arbitrary number of custom drawers
- Drawers from bottom useful to be used as a kind of "dialog"



OverlaySheet

Callery > Buttons
Open Bottom drawer
Open Sheet
Toggle Action Button
Show Passive Notification
Descine Notification Action

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam id risus id augue euismod accumsan. Nunc vestibulum placerat bibendum. Morbi commodo auctor varius. Donec molestie euismod ultrices. Sed facilisis augue nec eros auctor, vitae mattis quam rhoncus. Nam ut erat diam. Curabitur iaculis accumsan magna, eget fermentum massa scelerisque eu. Cras elementum erat non erat euismod accumsan. Vestibulum ac mi sed dui finibus pulvinar. Vivamus dictum, leo sed lobortis porttitor, nisl magna faucibus orci, sit amet euismod arcu elit eget est. Duis et vehicula nibh. In arcu sapien, laoreet sit amet porttitor non, rhoncus vel magna. Suspendisse imperdiet consectetur est nec ornare. Pellentesque bibendum sapien at erat efficitur vehicula. Morbi sed porta nibh. Vestibulum ut urna ut dolor sagittis mattis.

Morbi dictum, sapien at maximus pulvinar, sapien metus condimentum magna, quis lobortis nisi dui mollis turpis. Aliquam sit amet scelerisque dui. In sit amet tellus placerat, condimentum enim s lobortis finibus. Susp blandit. Nullam volutions quantum energy rerit aliquam.

Donac non-com at diam nosuoro convallis. Vivamus ut cor

Better than drawers, to have big, scrolling dialog-like pages that can be dismissed by gesture, use OverlaySheet

ScrollingPage

- Many Pages on mobile devices are scrollable controls, either lists of items or just complicated layouts that don't fit in the screen
- ScrollingPage manages that without the need of explicitly including Flickables
- Supports the popular "pull down to refresh" gesture present in many mobile apps
- By overpulling, the whole UI gets dragged down making it reachable by thumb

Collery > Long List	view	
C		
Item 0		
Item 1		
Item 2		
Item 3		
Item 4		
Item 5		
Item 6		
Item 7		
Item 8		
Item 9	\langle Gallery \rangle Buttons	
Item 10	Button	!!</td
Item 11	CheckBox	<⊞
Item 12	Radio Button	(!!!
	Progress Bar	(!!!
	Slider	<⊞
	Switch	!!</td
	Text Field	(11)

ListItemWithActions

< Gallery

Button	(!!!
CheckBox	!!</td
Radio Button	<
Progress Bar	<
	€ ⊠
Switch	!!</td
<t field<="" td=""><td><Ⅲ 3</td></t>	<Ⅲ 3
Multiple Columns	<
List View	<
Non Scrollable Page	!!</td
Colors	!!</td

÷

 A list item with a standard look, plus a draggable handle to reveal actions





- The application is divided in "pages"
- The class sets a standard behavior
- Exposes "actions" properties
 - main: main Primary Action Button action
 - left: optional smaller button at left of Primary Action Button
 - Right: optional smaller button at right of Primary Action Button
 - contextualActions: list that will go in the contextual drawer



ApplicationHeader



- The top of the application is reserved to a "title" area that acts as a breadcrumb in the currently open page hierarchy
- Content can be customized using
 AbstractapplicationHeader instead



- Action: "model" representation of an action
- ListItem/BasicListItem
- Heading
- IconGrid
- Label
- SplitDrawer



Success Story

- SubSurface
- Dive log app written by Linus Torvalds, Dirk Hohndel and others
- Desktop app
- Used to be GTK+, migrated to Qt
- Started an Android version in Qt sharing most of code
- Early adopter for Kirigami, good feedback so far



SubSurface

Subsurface	
	sab gen 1, 2011 8:00 PM #12
	sab gen 1, 2011 6:00 PM #10
Cloud credentials	
Preferences	
Manage dives	$\left.\right>$
GPS	\geq
Developer	\geq
Run location service	

<u>12th te</u>	est dive, 3 tanks, 3 tank
Dive 1	2
Date:	sab gen 1, 2011 8:00 PM
Location:	12th test dive, 3 tanks, 3 tank changes
Use current GPS location:	
Depth:	30m
Duration:	30min
Air Temp:	27.0°C
Water Temp:	26.0°C
Suit:	
Buddy:	
Dive Master:	
Notes:	Shows three pressure plots, each from 200 to 100bar and gaschange events at 10, 20 and 25 min mark. Last change is back to first tank.
	Save
	\bigcirc

SubSurface

\equiv f_m Subsurface-mobile

Dive Log

12th test dive, 3 tanks, 3 tank changes	sab gen 1, 2011 8:00 PM
Depth: 30m Duration: 30min	#12
10th test dive, 3 tanks, 1 tank change	sab gen 1, 2011 6:00 PM
Depth: 30m Duration: 30min	#10



Key takeaways

- Publicity (ehi, Linus is using Kirigami :p)
- It must be tier 1 (no dependencies besides Qt) to spark interest (makes some things more difficult for us)
- Many are going to be used *primarly* for Android applications, both by outside contributors *and* KDE people
- But encourages ports to Plasma Mobile

Future

- Smooth transition to QQC2:
 - pay attention to not have things that openly collide with them
 - Very useful stuff only in QQC1 (Action, ScrollView, StackView...) will still be used: probably depending from both for the time being
- Tier1 means not being able to use a lot of interesting kf5 stuff that have to be potentially reimplemented
- Packaging on mobile platforms can be tricky, Android is being fleshed out, iOS still at very early stages





- HIG: https://community.kde.org/KDE_Visual_Design_Group/KirigamiHIG
- API: https://api.kde.org/playground-api/libs-apidocs/kirigami/html/index.html
- Gallery on Android: https://play.google.com/store/apps/details?id=org.kde.kirigamigallery
- IRC: #plasma on freenode
- Mailing list: plasma-devel@kde.org