Multiview Plugin "Tour Guide"

Overview:

Multiview allows the use of panels inside JQM pages.

Panels are basically page containers (viewports) that contain nested JQM pages. Nested pages can be on board when the panel loads or loaded in externally on demand.

The whole plugin resides inside a regular JQM page (= wrapper page). A user can either swap the wrapper page's content section for any number of panels or add panels after the content section.

The plugin allows headers and footers to be global across all nested pages (if put inside the wrapper page) or local (if put inside any nested page).

Using multiview it's easy to set up splitscreen layouts (menu/main panel), enable "true" multipages (fullwidth panel, nested pages, global header/footer) and drop any number of popover panels on a given JQM or multiview page.

The plugin is not finished yet. The basic functioanlities all work, but integrating with JQM and ensuring everything works with more than one wrapper page in the DOM is challenging...

Demo:

Demo here: http://www.stokkers.mobi/valuables/multiview/show/index.html

Features: http://www.stokkers.mobi/valuables/multiview/show/index.html#demosHeadliner

Files on Github: https://github.com/frequent/multiview

Quick Setup:

To use the plugin use the multiview.js, multiview.css plus my modified JQM1.0multiview.js.

Add *data-wrapper="true"* to a JQM page to fire the plugin. Replace content section in this page with panels (type can be: main/menu/fullwidth/popover) or drop panels behind the content section (popover).

Panels have to be divs and need attributes *data-panel="menu/main/popover/fullwidth" data-id="your-panel-name" data-role="panel"*. Most important optional attribute *data-hash="history"*

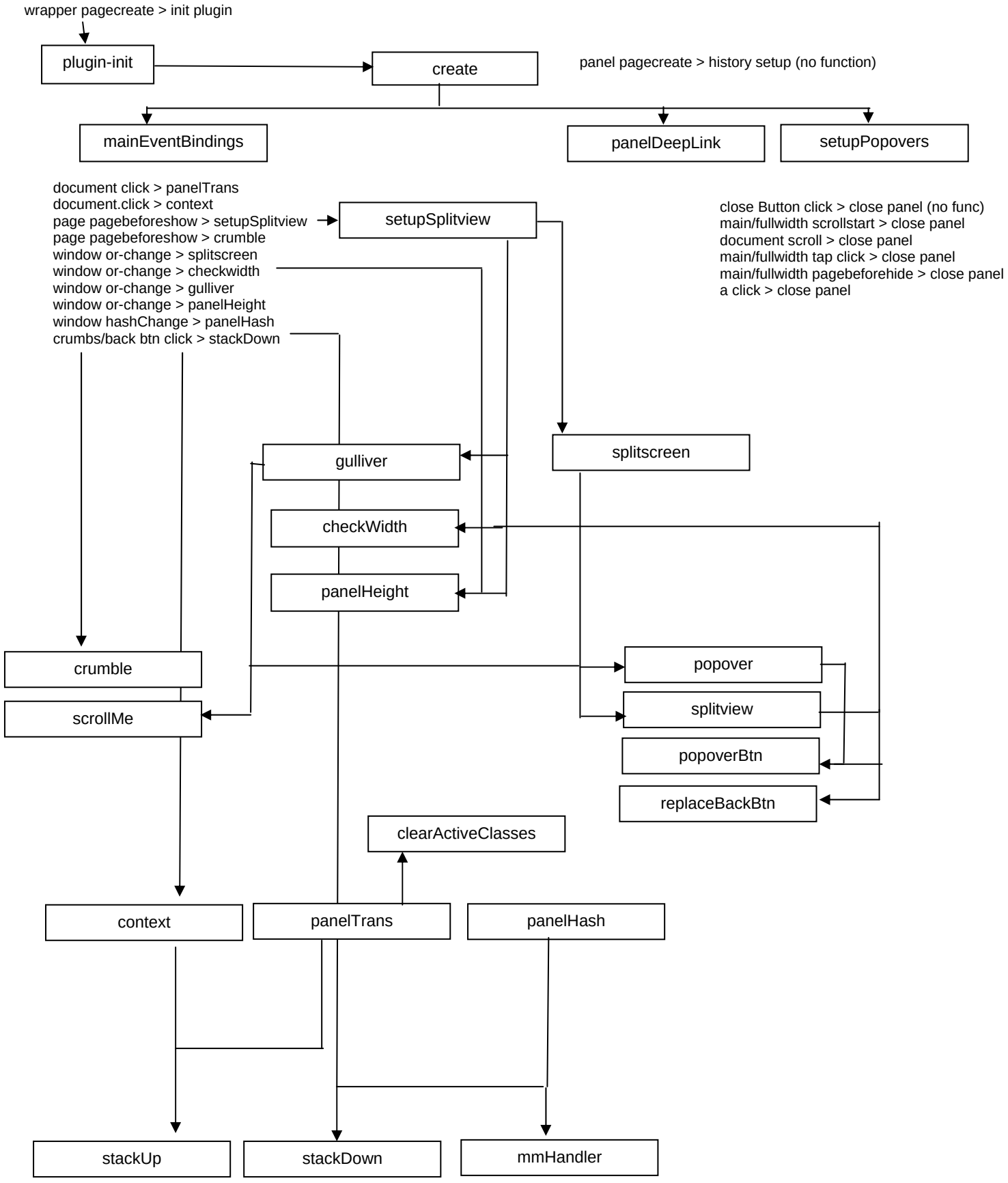
Drop any number of nested JQM pages inside a panel. Be sure to set *data-show="first"* to the page you want to see at init.

To set up panel transitions make sure to add *data-target="panel-name"* to a link. To create a button to open a popover, just add *data-panel="panel-id"* and a class of *. toggle_popover* to the button.

That should get you more or less going.

Function Map:

This map shows what functions call what. The next section briefly describes all plugin functions. I hope you can find your way around.



A	function name	description
		Setup
A1	options	options
A2	vars	flags used throughout the plugin
A3	create	setup needed once
A4	setupSplitview	setup needed every time a new multiview page is added to DOM
A5	setupPopovers	setup popover panels
		Screen Modes (splitview/popover/fullwidth)
A6	crumble	panel history back buttons
A7	replaceBackBtn	panel back buttons
A8	popover	set screen to popover mode
A9	splitview	set screen to splitview mode
A10	popoverBtn	insert toggle menu button
A11	splitscreen	set if popover or splitview mode
		misc
A12	context	manage context loading
A13	scrollMe	selectively init scrollview
[A14]	findClosestLink	same as JQM
[A15]	getClosestBaseUrl	same as JQM
[A16]	removeActiveLinkClass	same as JQM
		panel and page formatting
A17	gulliver	changes depending on screen mode
A18	checkWidth	adjust width depending on splitview/popover view
A19	panelHeight	set panel = viewport height (mimic body viewport)
A20	framer	determine screen mode (fullscreen/regular)
		panel history
A21	stackUp	add entrys to panel history
A22	stackDown	remove entrys to panel history
A23	clearActiveClasses	clear active button classes
		panel navigation
A24	panelTrans	handles panel transitions
A25	panelHash	handles panel hashchanges
A26	panelDeepLink	handles panel deeplinks
		misc2
A27	mmHandler	determines which panel to hashChange on
A28	mainEventBindings	event bindings for the plugin

A1 options

include:

switchable options	allow to hide/show the menu panel in splitscreen mode
menu options	menu default width, menu button default styles
external sitemap	object to carry #pageID / #panelID for all pages that could be loaded into panels externally to allow deeplinks to pages not "on board" when the wrapper page loads
fixed element margin	top and bottom margin for ui-fixed-element-top/bottom
thresh	threshhold page width to switch between splitscreen and popover mode

Todos:

- Haven't looked into deeplinking external panel pages. The sitemap object is just an idea
- fixed-element margins not optioned > fixed-elements behave like fixed header/footer
- Haven't made threshold width as option
- Need to add fullscreen mode option

A2 vars/flags

Needed to make navigation work. Took the idea from JQM - $ignoreNextHashChange

Currently used:

$ignoreNextHashChange	same behavior as JQM $ignoreNextHashChange
$crumbsBlockStackUp	blocks panel transition when a panel back button is clicked
$windowHistoryCounter	keep counter for every panel transition/hashChange (currently not used)
$hashJoker	needed for backwards transition to block hashChange
$contextBlockNextHashChange	block 2nd hashChange event from context transition (2 panel transitions)
$jqmHistoryStackAtInit	remember jqmHistoryStack when doing the first panel transition in order for JQM to take over again

A3 create setup once

The plugin fires when the first JQM page with *jqmData(wrapper="true")* is created. Afterwards I lock up the live binding to prevent the plugin from firing again (code is at end of plugin)

Key in create is a live("pagecreate", ...) listener for new panels which have a history attribute set by the user (data-hash="history" or data-hash="crumbs"). The binding is nonsense (pagecreate on a panel) but it works, so I'm not asking why.

A4 setupSplitview setup needed every time a new multiview page is added to DOM

This should fire every time a new wrapper page is loaded into the DOM. It enhances the first page on every panel (Kinblas), manually adds data-url and data-internal-page to all panel pages and fires splitscreen and screen formatting functions.

A5 setupPopovers setup popover panels

includes the full popover functionality, adds optional popover triangels and makes sure, popovers are hidden at pageload unless data-show="once" is specified.

The two keys for popovers are hideAllPanels() and the toggle-button click behavior.

There are 9 different "user interactions" which close popovers - from straightforward clicking the close button or tapping on the background panel to a cross panel pageChange, which on smartphones needs to close the popover if the changePage was initiated from it.

Popovers have rules:
- only one can be open at a time, otherwise what to show on smartphones
- closing a popover resets it to page with data-show="first"
- closing a popover resets it's history to base level = panel stack-height = 1
- in fullscreen mode (smartphones) popovers look like regular JQM pages

Popovers' dimensions are set by .ui-popover, position can be set by user.

Popovers need to be ui.element-fixed-top (= behave like fixed header) or .ui-element-fixed-bottom (= behave like fixed footer) to make them scroll along when the user scrolls.

Todos:
- I'm using history.replaceState here, because when the popover closes, the URL is "stuck" at the last page active in the popover before closing. Not working very nice
- I wanted to also hide popovers on orientationchange when not in fullscreen mode, but this strangely causes problems with deeplinks = breaks them, so commented out for now.

A6 crumble panel history back buttons

If a panel has an acitve history (*data-hash="history/crumbs"*), this function drops back buttons on every page transitioned to.

The ticky part is to find the page and header (wrapper vs. panel page, global vs. local header) and "what" is on the left side. Left can be:

a) nothing or back button	= remove and drop a new back button
b) some other button/select/other element	= replace it with a controlgroup inside a button wrapper
c) a button-wrapper with left button(s)	= replace first button with controlgroup
d) a button-wrapper with left controlgroup	= pick first button inside controlgroup that's not a back button, make a new controlgroup and replace the current controlgroup

The plugin uses *.ui-headWrapLeft* and *.ui-headWrapRight*, which are basically wrappers, which can contain multiple buttons and controlgroups and align them nicely inside a header. WrapLeft is positioned using *.ui-btn-left*, wrap right uses *.ui-btn-right*

Todos:
- works nicely but I think this is too customized for a generic plugin. Although pretty easy to use.
- fires with every pageBeforeShow, not sure if this is nice or not. Lags on Window Mobile.

A7 replaceBackBtn panel back buttons

Removes all menu buttons when switching from popover mode to splitscreen mode. Fired as cleanup from splitview function.

A8 popover set screen to popover mode

Set up popover mode = main panel spans across full screenwidth and the menu can be toggled as a popover using a menu_toggle button. Function adds/removes necessary classes and is fired from splitscreen function.

A9 splitview set screen to splitview mode

Same as popover function = add/remove necessary classes. Also fired from splitscreen

A10 popoverBtn insert toggle menu button

Adds the menu toggle button. This is complex because it has a lot of options, plus if the switchable option is active it needs to be dropped where-ever specified. If the button should go inside the header and there is already stuff on the left side, it will be dropped inside a button wrap after the first element (supposedly a back button) or merged into wrapped controlgroup with an existing button.

Todo:
- also too much customization

A11 splitscreen set if popover or splitview mode

Determines whether to fire splitview or popover depending on orientation and screen width

A12 context manage context loading

The context function allows to do "double transitions" (like load a menu and corresponding main page"). It currently can only be fired from a link with specified *data-context-panel* and *data-context(-page)*. The plugin fires a 2nd panel pageChange. Needs $contextBlockNextHashChange to block 2nd hashChange.

Todos:
- only works menu>main, all other variants don't work. I guess because of pageContainer not being set correctly?
- cannot be called programmatically

A13 scrollMe selectively init scrollview

Inits scrollview. This is only needed in splitscreen mode for popovers and the menu as well as in popover mode for popover panels. Everything else should be done by native scrolling.

Todos:
- not inside the plugin but if you add form sliders to a panel using scrollview the bugfix #2072 currently blocks all other panels from scrolling.

[A14] findClosestLink same as JQM
[A15] getClosestBaseUrl same as JQM
[A16] removeActiveLinkClass same as JQM

Would save same code if I could call these form outside JQM.

A17 gulliver changes depending on screen mode

this sets up fullscreen mode = all panels "become" full screen pages. It is triggered if page width is below 320px (should be an option) or if a popover panel is larger than screen height.

In fullscreen-mode popovers loose all panel CSS, close buttons become back buttons etc, so it more or less feels like real pages.

I'm also changing iconpos to notext on all buttons labelled with class .ui-iconposSwitcher-a/div. Saves space in the header and this way you can fit 4 buttons on each side if you wanted.

A18 checkWidth adjust width depending on splitview/popover view

As the menu can have width and min-width, the main section needs to adjust width on resize to avoid panels overlapping. Also, as main/menu panels have pos:relative/static I also need to set the CSS of nested panel pages. I have not found another way to do this nicely.

Todos:

- not necessary, if the menu would be fixed at xyz px.
- see if panel width can be set and inherited to panel pages without breaking transitions and keeping panel-css:pos="static" or "relative" to mimic body-viewport. Not easy.

A19 panelHeight set panel = viewport height (mimic body viewport)

Sets panel (= viewport) height to screenHeight less global header/footer.

A20 framer determine screen mode (fullscreen/regular)

Sets screen mode to small, medium or large (for gulliver function)

A21 stackUp add entrys to panel history

Adds new entries to the respective panel history stack, increases $windowHistoryCounter and keeps JQM history at $jqmHistoryStackAtInit .

Also sets $('html').data({'backAtBase':false}); to false, because panel base level is left. Sets a last stand $('html').data("lastStand", "standing"); so JQM does not take over once panel-base is reached again (dropping lastStand), but only on the following transition.

A22 stackDown remove entrys to panel history

Removes last entry from the respective panel history stack, increases $windowHistoryCounter keeps JQM history at $jqmHistoryStackAtInit panel and set a flag once all panels are back at their inital (base) level, so JQM can take back over with the next non-panel hashChange.

A23 clearActiveClasses clear active button classes

clears active buttons on all panel transitions

A24 panelTrans handles panel transitions

This has a lot of duplicate code from JQM.

Basically this checks for links with a specified target panel = pageContainer option and then does a changePage either on the panel or cross panel. The navigation part is similar to the splitview plugin by Asyraf.

After the transition, stackUp is fired to add new entries to the panel history.

All the flags are also set after the transition = $crumbsBlockStackUp blocks new entries into history on crumbs/back button transitions, $ignoreMyOwnNextHashChange blocks hashChange firing after the transition, $hashJoker to make sure a panel is transition fired, which I can check in hashChange to determine where the hashChange is coming from.

Todo:
- get to work without duplicate code
- cannot be called programmatically
- pushState not really working

A25 panelHash handles panel hashChanges

This function handles panel backwars transitions.

The history works like this:
- on every panel transition the transitioned-to panel gets a new entry (if history is active)
- entries (so far) only include hashes
- main and menu panel are linked and increase/decrease together
- when main gets a new entry, menu gets a "yield" entry and vice verca
- as popover panels reset when closed, their history always goes down first

On every hashChange this function searches for the highest panel history stack and creates an array with all highest stacks. So if there are 4 panels and they are all in base setup = stack height =1, the longest array will include 4 elements = each panel stack object with it's entries.

The plugin first checks for open popovers. If no popovers are open it looks for the highest stack to decrease or if menu/main panel for the last entry not being yield (done by mmHandler function).

This ways the page for the backward transition is set.

The 2nd part (no "to" defined") bloats up this function, because I need the whole "to" logic again. The problem is when the wrapper page initially loads, the URL does not show every panels data-show="first" page. So the URL is page.html instead of for example page.html#firstpanelpage. Therefore on the last transition to the #firstpanelpage the hash is empty = no "to" defined, when in fact there is a page to go. To override this, I need to use the "to" part again.

After the backwards transition is made, stackDown is fired and the flags are set appropriately.

Todo:
- get to work without duplicate code
- to much code in the latter part. Could be done easier I think

A26 panelDeepLink handles panel deeplinks

This enables deeplinks to all pages that are "on board" when the wrapper page loads.

Todo:
- figure out a way to enable deeplinks to panel pages that will be pulled in later.

A27 mmHandler determines which panel to hashChange on

Checks menu and main panel to determine the panel page for backwards transition

A28 mainEventBindings event bindings for the plugin

Contains the main event bindings.

The pageshow binding was hard, because this fires a-plenty on a wrapper page with many nested pages and I need to make sure new wrapper pages pulled into the DOM would also get the plugin treatment without the plugin auto-firing away.

B	**Changes to JQM**	
B1	**core utilities**	**convertUrlToDataUrl - add check for wrapper and panels**
B2	**transitionPages**	**block scrollTop on popover panel transitions**
B3	**transitionPages**	**block hide event on panel transitions to not drop wrapper page**
B4	**removeActiveClass**	**add clauses to remove active page on wrapper/page**
B5	**changePage**	**check for target and preventDefault**
B6	**_handleHashChange**	**check for panels and preventDefault**
B7	**pushStateHandler**	**check for panel transition and modify href accordingly**
B8	**fixed toolbars**	**add support for global toolbars and element fixed top/bottom**
B9	**init**	**add clause to always append loader on body and not panel**
B10	**init**	**add clause to enable panel page deep linking**

To easily see what I changed in JQM search for "XXX FREQUENT". I labelled every change with this tag.

B1 core utilities convertUrlToDataUrl - add check for wrapper and panels

need to add a check, which enables loading external pages into panels of a wrapper page, which was added to the DOM as second aka data-external="true" page

B2 transitionPages block scrollTop on popover panel transitions

block scrolltop on transitions inside a popover. Best look at non-blinking transitions you will get.

B3 transitionPages block hide event on panel transitions to not drop wrapper page

check for data-internal-page="true" to block page hide on panel transitions (would remove wrapper from DOM)

B4 removeActiveClass add clauses to remove active page on wrapper/page

drop active class on wrapper or keep it depending on transition - this was complicated to fix

B5 changePage check for target and preventDefault

add pageContainer as new option, preventDefault if panel transition

B6 _handleHashChange check for panels and preventDefault

not nice. block JQM on panel backwards transition and make sure JQM takes over again ta the correct moment

B7 pushStateHandler check for panel transition and modify href accordingly

add if-clauses to also set href depending on interanl or external wrapper page.

B8 fixed toolbars add support for global toolbars and element fixed top/bottom

accomodate global/local toolbars, fixed element top/bottom. I have improved this, but it's still slow...

B9 init add clause to always append loader on body and not panel

B10 init add clause to enable panel page deep linking
tweak for deeplinks