header

main application header

main application toolbar

p.l p.l

50u

p.s

page width

the arrow is visible only when the view is nested, main views don't have back arrows to go to home

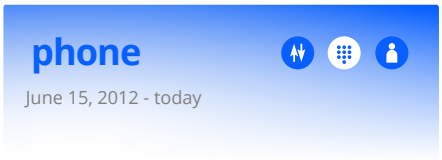
tabbed view header

views slide in with parallax with the header so that when the title reaches the center of the header, the view is also centered in the screen.

If the view has large canvas, switching views can be done by swiping on the header or clicking the desired tab.

tabs should have an `icon` property, for compatibility with the Breeze theme, even if it's not visible here.

the tabbed view in breeze theme



If the toolbar icons are more than 3 or the app requires a menu, 3 dots signal that the header can be dragged down to reveal a menu

The dots are smaller than comfortable to tap, if tapped the menu should open anyway, but for easier operation, the user can just swipe down anywhere on the header, even if it's over a button

l.l

header-gradient.png

icon-triangle-left.png

24pt/17u

bold (75)

#ffffff

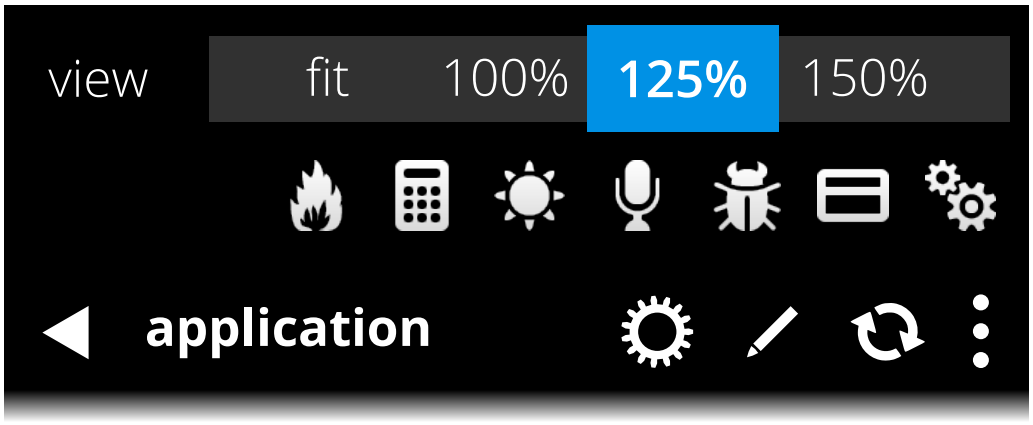
normal (50)

light (25)

dots-vertical.png

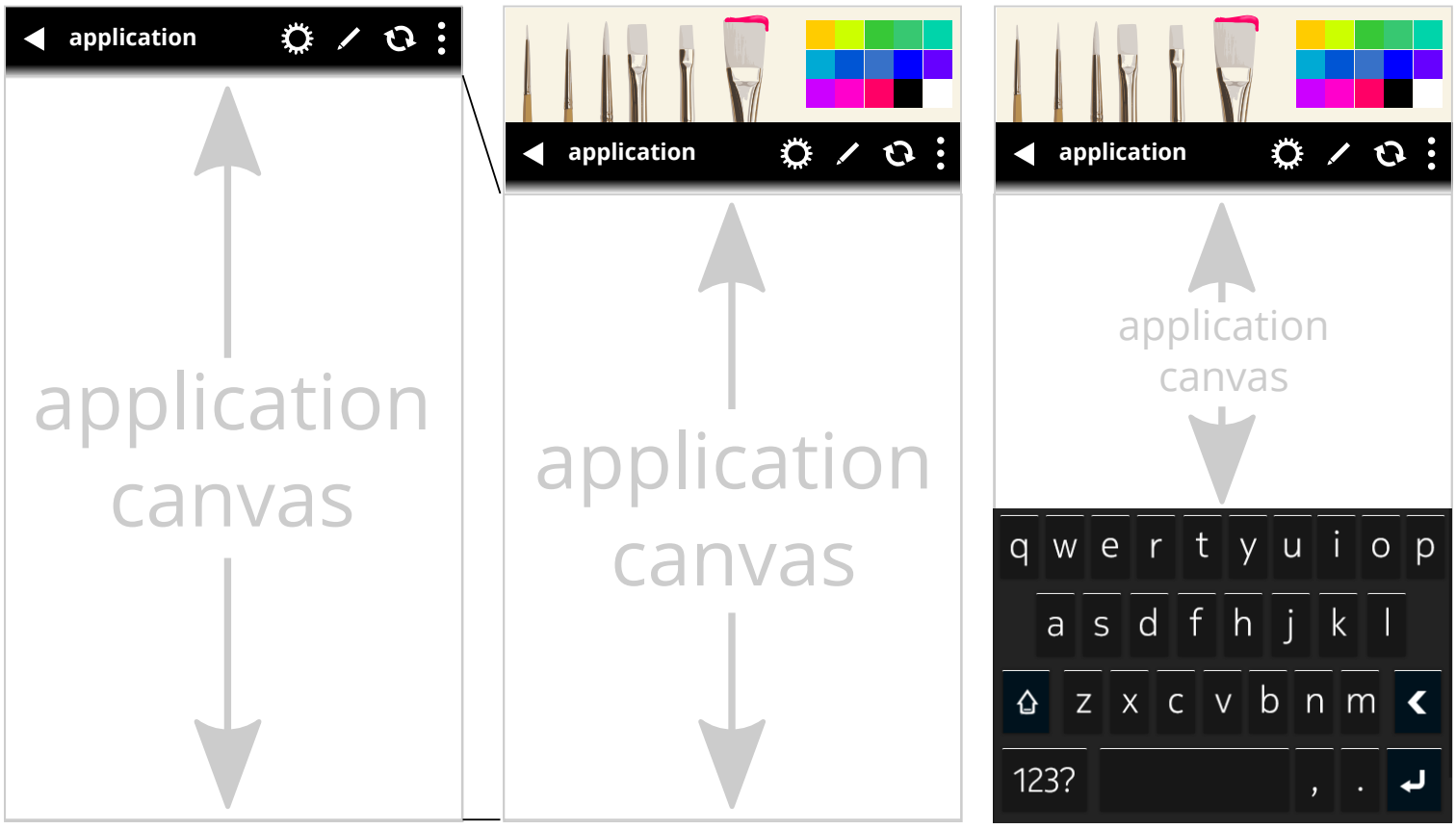
header, continued

it is important that we make most of the available space, so the header-menu shouldn't be strictly a list view. (I used the *gentleface* icons for display purposes. Our set of monochrome action icons is TBD.



we also want the menu to be non-modal, i.e. allow the user to continue working on his app (let's say a drawing/sketch app) while being able to change colors)

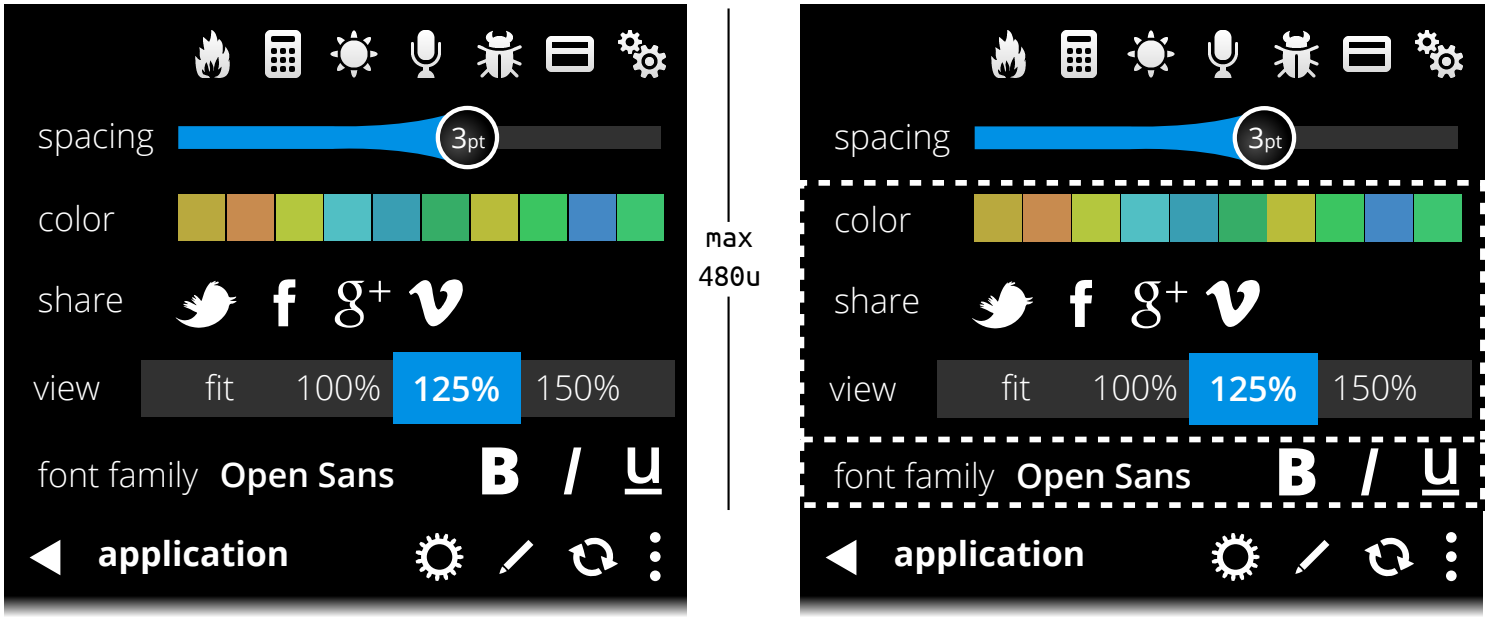
This means that the menu will not overlay the application, nor push it down off-screen. It will resize the available space of the page, just like the virtual keyboard does.



There should be an api so that the developer will be able to auto-close the menu in case the vkb pops up, as the remaining space is too small.

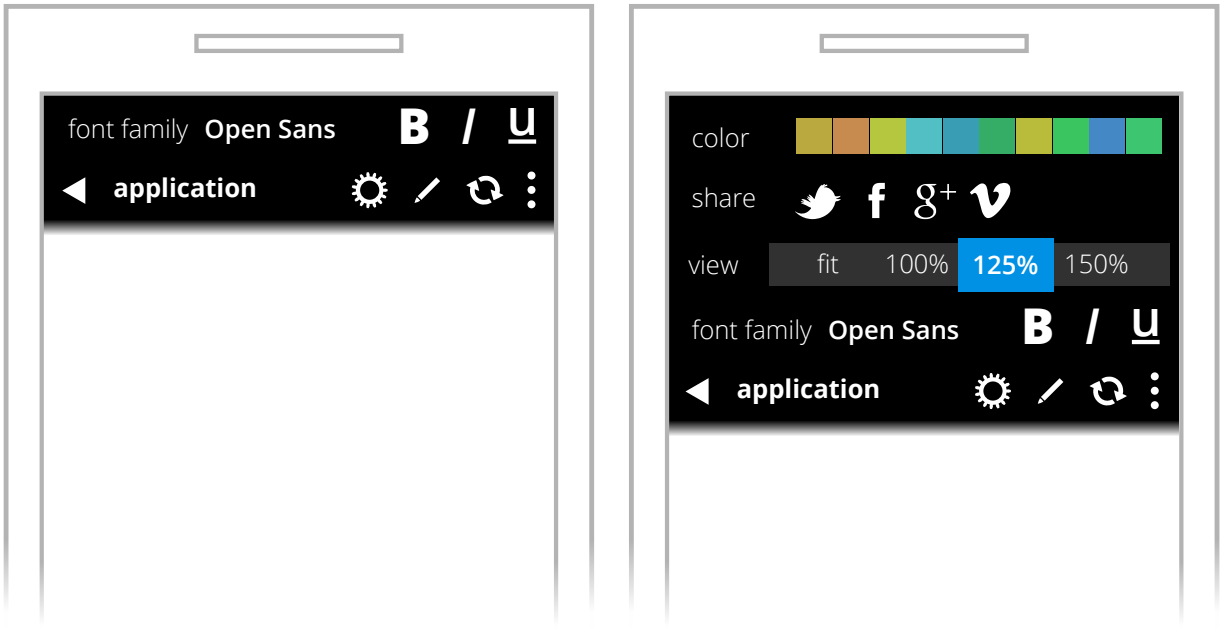
header states

Let's say we have a very big application, full of controls. The header might be something like this:



The user might want only a handful of controls visible all the time, the most useful ones. The header dock should have multiple open states, defined by invisible levels. (dashed line)

This would allow the following states additionally to a fully open dock. The levels inside the dock are not visible, but can be perceived by the user using speedbumps.



For this reason, it is advised that the most frequently used items are positioned towards the bottom of the dock.

Header toolButton pressed effect

header in landscape

l.m p.m

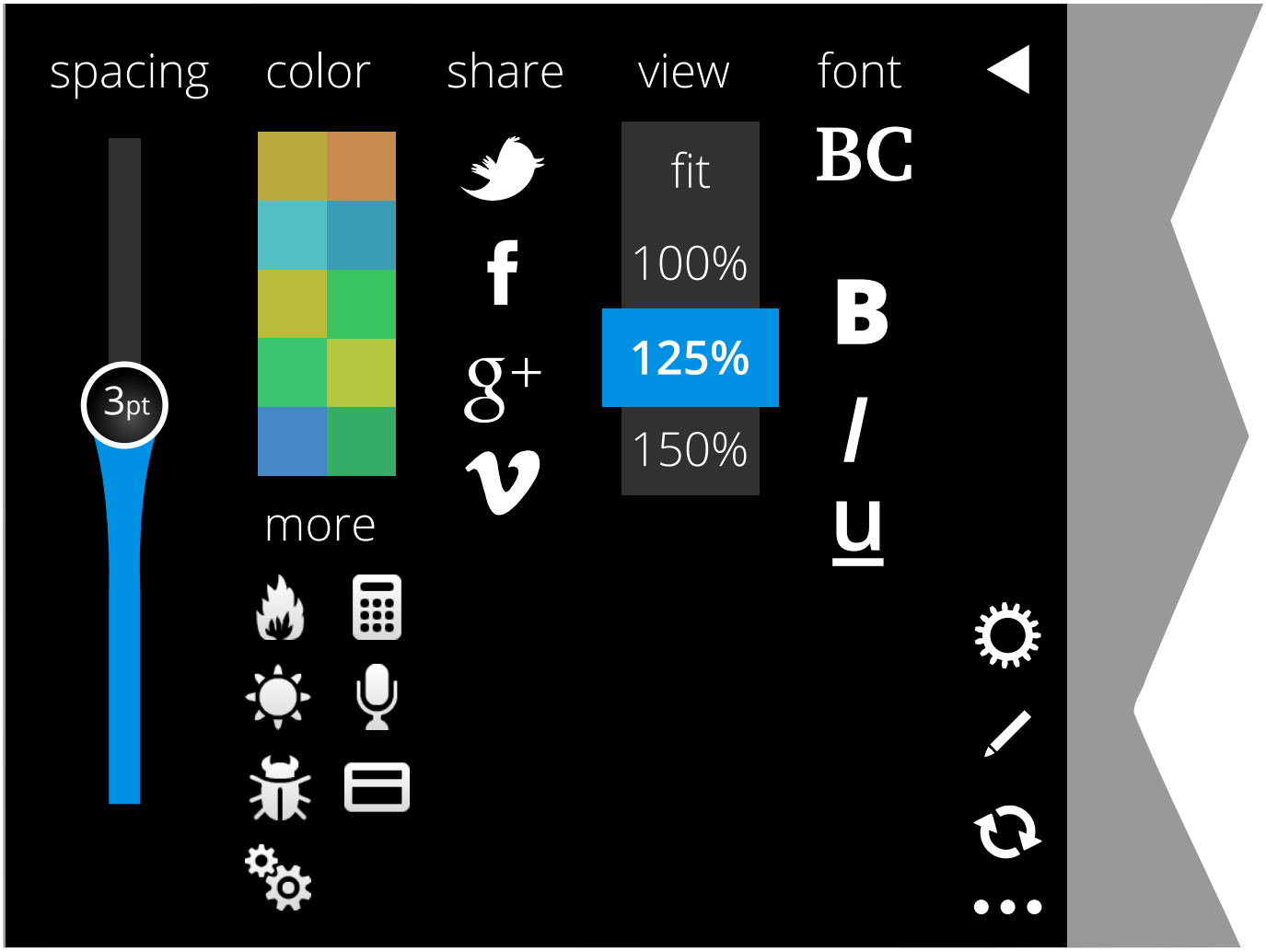
If the virtual keyboard pops up while the side drawer is open, or if the developer puts too many items in the drawer that do not fit, the excess is clipped.

Levels orientation

If the developer has opted to use Levels in both orientations he must arrange his controls in a way that fit properly in tall columns and wide rows and provide width and height for the Levels. If both dimensions are configured widths will be ignored in portrait and heights will be ignored in landscape.

If only one of the dimensions is configured, the levels will work as expected in the orientation that makes sense (height makes sense in portrait and width in landscape). In the other orientation only fully open and fully closed orientation will be allowed, and the contents of the dock will be exactly the same in both orientations. (The dock area is square in fully open position allowing this)

480u



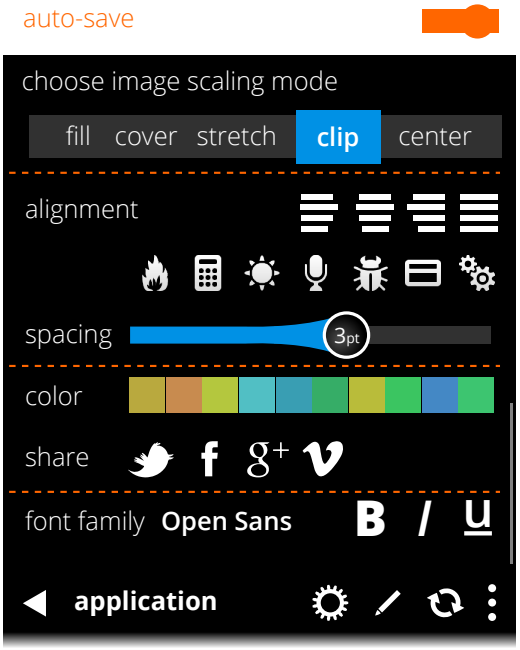
If the virtual keyboard pops up while the side drawer is open the drawer is automatically closed.

header with too many children

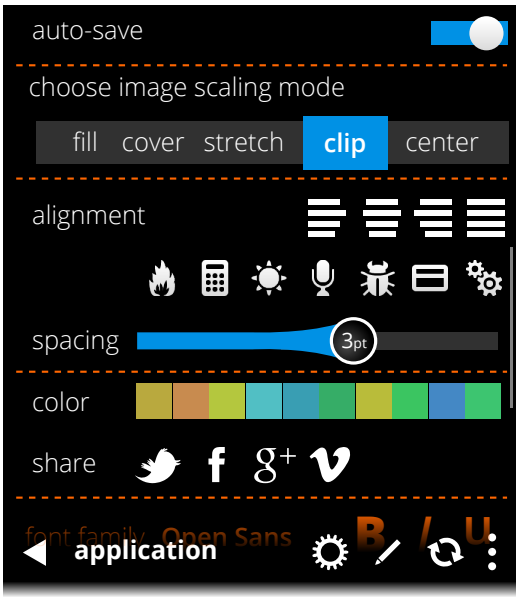
When there are more controls to place in the dock than they fit in one screen, the user can scroll perpendicularly to the levels (if levels are horizontal, user can scroll vertically)

The scrolling action snaps to levels.

Let's assume for the following example that there are more controls hidden above *scaling mode*



The orange items (elements outside the element and dashed lines) are not visible in the UI, they indicate the overflowing elements and the boundaries of levels respectively.



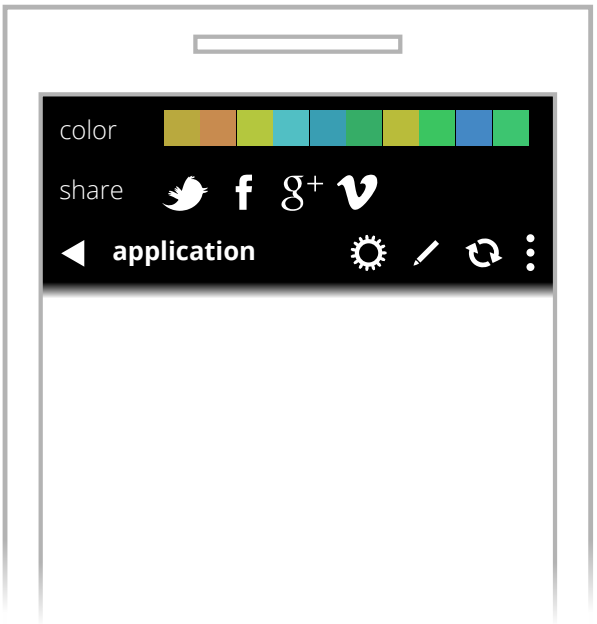
This is a scrolled down view (notice the scroll indicator). A new option appeared above. The bottom level cannot be clipped halfway (the top one can). The view snaps in a way that the bottom is always a level boundary.

In this state the user can again close the dock either fully or partially so that any number of levels is visible. Speedbumps continue to work. It also enables the user to show only the 2nd or only the 3rd level.

The same thing works for landscape view, with tall column levels.

In landscape view with wide row levels, the view still scrolls vertically, the view still snaps so that the bottom line is always aligned with a level boundary, but there is no possibility of partially closing the dock.

The scroll indicator and other controls in this view are indicative, and not perfectly positioned



Scrolling through levels still works when the dock is halfway closed

Again the scrolling snaps to the bottom of the dock. If a level bigger/ smaller than the view is scrolled into view, after the user releases his finger the dock slides open/closed until only that level is visible.