connect IQ™

powered by Garmin

USER EXPERIENCE GUIDE

DESIGNING FOR CONNECT IQ

The Connect IQ system allows third party developers to bring their design and experience to Garmin products. Your ideas and content are what will power the wearable revolution!

This document will go over some of the guidelines of designing an app for a Garmin wearable device.

THINKING WEARABLE

The KISS principal[1] is one of the best rules of thumb when making a wearable app. One of the synonyms of the word “watch” is “to observe”. When designing smart watch apps, it is best to keep in mind that watches are best when used to observe. Great wearable apps stay out of the user’s way, and keep the interactions to a minimum. Give the user fast access to the information and key interactions they care about. Garmin devices have access to a number of different sensors, including GPS, compass, accelerometer, barometer, as well as the ANT+ network. Let the user see information about themselves in interesting ways.

[1] Keep It Simple Stupid. Also accepted: “Kiss the fuzzy monkey who wuvs you so much”

DESIGN FOR APP TYPES

The Connect IQ app types are based on the use case of the app. Watch apps have full control of the device. Widgets provide glanceable information to the user. Data fields allow customization of the Garmin activities. Watch faces provide custom displays for the home screen of the watch.

In this section we will delve into the specific guidelines for each use case.

WATCH APPS

Watch apps are by far the most robust type of app available. These allow the most flexibility and customization to the app designer. They also provide the most access to the capabilities of the wearable device, such as accessing ANT+ sensors, the accelerometer and reading/recording FIT files.

The suite of Garmin wearables are each designed to fulfill different needs and behaviors of active individuals, from endurance runners to triathletes to outdoor enthusiasts and adventurers. The core focus of these wearables centers on the recording and tracking of activities, from running to hiking to skiing. Users of the different Garmin wearables desire to track specific types of data and great care should be taken in designing your watch app to understand the needs of the user doing a particular activity or task and provide appropriate feedback, metrics and configurability to give the user the best experience.

- **Next/Previous**: typically moves through a page loop
- **Lap**: used to mark lap or split times within the activity
- **Back**: must be available when the activity is not recording. Back behavior is not typically available while an activity is being recorded. If the app is structured in a page-based manner similar to the Garmin activities, users will already be accustomed to not having a back function during the activity.
- **Menu**: Handle the menu behavior, present them with a menu of options.

The initial view of the app should be a call to action. If your app represents some form of activity like hiking or weight lifting, the initial view of the app should ask to be started. Present the user with information from the sensors that make them want to hit the start button.

Garmin commonly uses page loops to present multiple pages of information. Page loops are a carousel of pages, each one a unique view on the activity. This is a common metaphor in Garmin products and easy to implement in Connect IQ.

When your app is presenting large amounts of text to the user, try to keep information in the center of the screen. On round screens the top and bottom of the screen provide a limited viewing area. Use the top for contextual headers, scroll arrows, and other small hints of information.

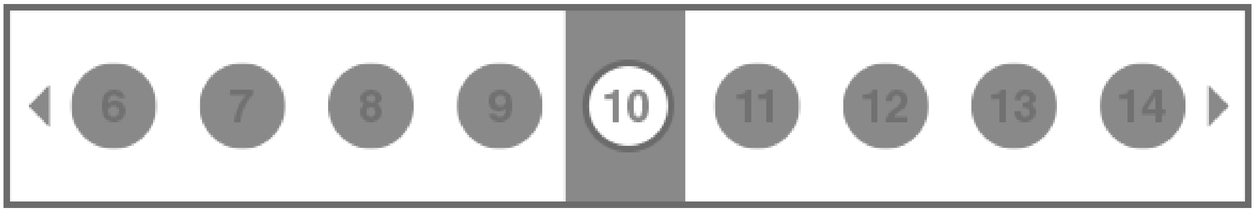
Finally, take advantage of the Connect IQ settings system. This lets the user customize your app through Garmin Connect Mobile.

WIDGETS

Widgets provide glanceable and relevant information accessible from the watch face. The most important element of the widget is the main page. This is what the user will see upon navigating to your widget; for the best experience this main page should provide relevant, timely and easily digestible information to the user.

The base view of a widget should be a presentation of critical information. Put key information the user will care about on the main page. The main page should only handle the launch widget behavior and the menu behavior. If the user performs the launch widget behavior, you can push a page that presents deeper information.

Your widget should load and unload quickly. Widgets are presented to the user as a carousel, and they may be starting your widget only to get to the next one. If your widget takes too long to load or unload, users will remove it from the watch and punish your app with a one star review.



Because widgets are intended to be glanceable and live in a page loop, behaviors available from the main page are limited. Widgets can open a menu, and on touch screen products widgets can provide controls from the main page such as play and pause controls provided in a music player widget. To provide interaction on non touch products or to provide more views or content in your widget, it must be launched or entered to interact with it. If interaction is desired, widgets should either provide controls from the widget page or should be enterable, but these two widget interactions should not be combined. If more actions or configurability need to be supported, the widget can open a menu at any time when it is in view. Once entered, interactive widgets can be structured in a page-based manner, similar to watch apps, or hierarchically. Once a widget is opened, a back behavior must be provided to return the user to the main widget face.

Data Fields

Data fields can run inside an already supported activity on the device and are a great way to provide new metrics to users by performing calculations on data already being recorded. Because data fields are integrated in with existing activities, it is best if they appear in the same font and format as that used for the native data fields. For that reason, the simple layout is best as it will ensure your data field will have the same native look and will scale appropriately to all data screen layouts. If you would like to customize your data field, for example, by inserting a bitmap in place of a numerical value, you will need to ensure that your custom field will scale appropriately from a one field, two field, three field, etc. layout.

Use the Simple Data Fields when possible to guarantee your data field will have the native look and feel of the other Garmin Data Fields. Connect IQ will try to ensure your data displays with the best font and layout possible.

Watch Faces

Watch faces run continuously on the device and can have the most effect on power consumption. A poorly designed watch face – one that takes too long to draw - can greatly degrade the battery life of the wearable.

Because of power concerns, watch faces have the least access to APIs in the system. They have access to graphics, bitmaps, fonts, current activity tracker status, current battery status, and the user's activity profile. They cannot access the compass, GPS, or other sensors.

Watch faces should extend the `WatchFace` class in `WatchUi`. Watch faces update at once a minute in low power mode, or once a second when in full power mode. The watch will transition out of low power mode when the user gestures to look at it. When entering full power mode, watch faces call `onExitSleep()`, which can be used to trigger animations and timers. The `onEnterSleep()` method is called when entering low power mode and is used to terminate any active timers. Both of these methods are inherited from `Toybox.WatchUi.WatchFace`.

If you use custom fonts for numeric display, use the filter option to only load the critical glyphs. This will save memory that you can use for additional graphics.

Page Layout

Garmin makes purpose driven products, and the technologies of the product are chosen around the use case the product is intended. Making purpose driven products also influences the decisions of the screen technologies and fonts we choose for our products.

Text Fonts versus Number Fonts

Garmin devices have two kinds of fonts available – text fonts and number fonts. Text fonts are for displaying text and number. The text font will come in three sizes:

1. Small – chosen to be the smallest readable font on the display
2. Medium – larger than small; chosen to efficiently display text.
3. Large – larger than medium; chosen to be headers.

The number font is a larger size font that can only display number data. It has four sizes:

1. Mild – quarter screen display of numeric data
2. Medium – one third screen display of numeric data
3. Hot – one half screen display of numeric data.
4. Thai Hot – number covers most of the screen.[2]

Typeface

With the exception of the Forerunner 920xt[3], the text font for Connect IQ devices will be the Roboto Regular typeface. The number font will be chosen for the product based on the use case of the product.

User Fonts

Garmin tries to provide the best looking fonts for the device. Using the BMFont tool, you can import your own typeface into the product. Using the BMFont tool, you can import truetype font into a font usable by Connect IQ. There are some guidelines to follow:

1. Try to use the system fonts when possible, but use custom fonts for accent.
2. Large typefaces can take up quite a bit of limited resources. If you only need a typeface for a logo, it might be preferable to use a bitmap instead.

[2] Also, this font burns my tongue. Who am I kidding, the medium font burns my tongue

[3] Don't ask; it's a long story involving release dates and, surprisingly, Yetis.

3. If your font is limited to a set of characters you can use a filter when importing the font. This will save memory by not importing glyphs your application doesn't use. For example, if you want to have your own custom large number font, you can use the resource compiler to only include digits and necessary characters like colons[4].

Bitmaps

Garmin wearable products use displays that do not require a backlight. These displays use much less power than traditional cell phone displays, but will look muted compared to a cell phone display. The displays have two bits of red, green, and blue; 64 colors total.

Different wearable products can have different color depths. The least number of colors a product will have is 16. The RGB values of the available colors on the 16 color products are as follows:

```
//! White
COLOR_WHITE     = 0xFFFFFF
//! Light Gray
COLOR_LT_GRAY   = 0xAAAAAA
//! Dark Gray
COLOR_DK_GRAY   = 0x555555
//! Black
COLOR_BLACK     = 0x000000
//! Red
COLOR_RED       = 0xFF0000
//! Dark Red
COLOR_DK_RED    = 0xAA0000
//! Orange
COLOR_ORANGE    = 0xFF5500
//! Yellow
COLOR_YELLOW    = 0xFFAA00
//! Green
COLOR_GREEN     = 0x00FF00
//! Dark Green
COLOR_DK_GREEN  = 0x00AA00
//! Blue
COLOR_BLUE      = 0x00AAFF
//! Dark Blue
COLOR_DK_BLUE   = 0x0000FF
//! Purple
COLOR_PURPLE    = 0xAA00FF
//! Pink
COLOR_PINK      = 0xFF00FF
```

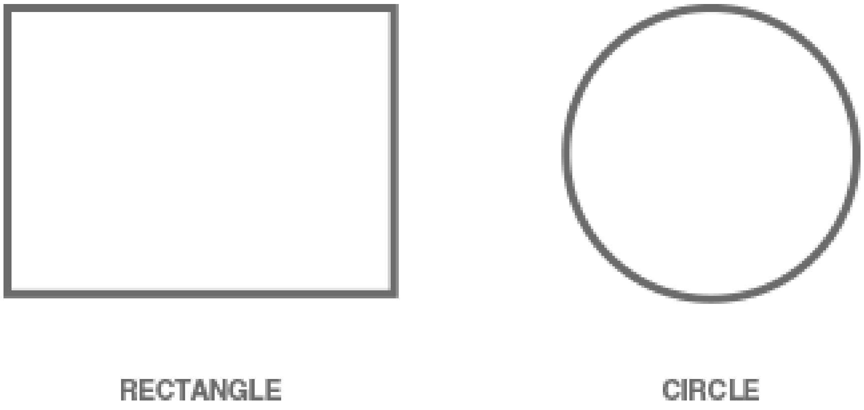
Using these colors for your bitmaps will make your app the most portable.

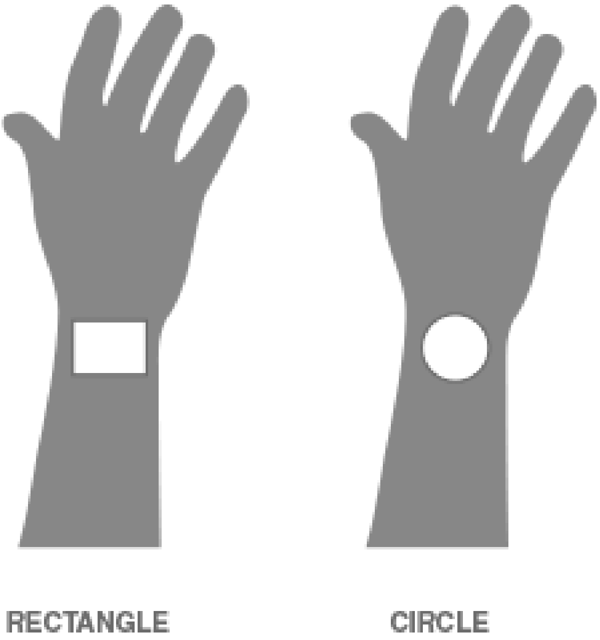
Try to use low bit-depth images when possible. More colors will use more memory, while one bit silhouettes can make for colorful user interfaces and saves memory for your application.

Page Layouts

Garmin makes products with many screen shapes.

[4] Fonts over 40 years old should have their colons checked every year by professionals.

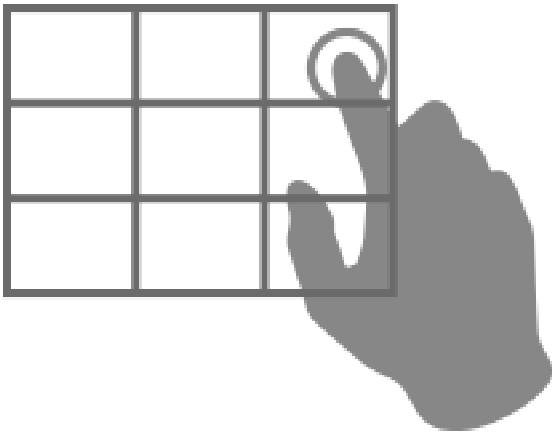




When designing your product, take full advantage of the layout system in the resource compiler and the simulator to test your layout. The resource compiler allows you to specify your page layout in XML and have it target a particular device. See the programmer's guide for more details.

Some, but not all, Garmin wearable products will have a touch screen. When designing a layout for a touchable screen, it is best to apply a rule of thirds: break the display up into a 3 x 3 grid and put your touchable elements clearly inside one of those grid areas. Using this rule ensure the user has enough finger space to select the touchable element[5].

[5] Certain products use a 3x3 touch screen that only has nine touchable areas, so consider it less than a guideline and more of a "do this or else".



When making a layout for a product without a touch screen, keep the user interaction to a one dimensional interface – up, down, enter, and back.

USER INTERACTION

Remember what I said earlier about limiting the amount of user input because a wearable app is a beautiful flower that should be seen and not heard yadda yadda yadda[6]? Clearly at some point the user has to interact with the app. Here are some rules for designing user interaction.

BEHAVIORS

Though the products have different styles of input, there are a number of common user behaviors between them. By working these behaviors into your design, you can make it much easier to port your product between different devices.

NEXT / PREVIOUS PAGE

Many of the Garmin activities are made up of page loops – information for the user spread across a carousel of pages. The next and previous behavior informs you that the user wants to go to the next or previous page.

MENU

The user wants to see a context specific menu.

ENTER

The user wants to select a highlighted item.

START / STOP

The start/stop button is used to begin timing/recording of an activity of some sort.

BACK / LAP

The user wants to back out of a page on the page loop. This is often multiplexed with the lap key, which is used to allow the user to demark a marker in their workout.

ENTER WIDGET

This is a new behavior specific to the widget app type.

BUILT IN DIALOGS

Connect IQ provides some basic Garmin dialogs for item selection, numeric entry, and confirmation. Using these widgets for basic operations will make it much easier to port your application across Garmin devices.

MENU

Menus let you present a list of items for the user to select from. They can be nested (a menu can launch another menu) for selection based flows. You can also use the resource compiler to define them. See the programmer's guide for details.

NUMBER PICKER

The number picker provides a UI for picking from a number of numeric value types. It provides an up/down/enter interface for selecting numeric values.

[6] Even I get tired of hearing me talk. I don't know how you can take it.

CONFIRMATION DIALOG

The confirmation dialog provides a simple yes/no dialog. This is useful when presenting a simple selection.

PROGRESS DIALOG

The progress dialog gives a standard wait dialog. It has two modes – one for progress where progress is measurable, the other is for showing a busyness without attaching a specific progress to it.

NOTIFICATIONS

If your application is monitoring sensor data, you may need to present the user with information about what you are monitoring.

If the notification to be presented is pertinent to the activity, present the information full screen. For example, if the user earns a badge in your game, this should be presented as a full screen page that times out back into your page flow.

If the notification is less pertinent to the activity (weak GPS signal), you can present a one line toast to the user. This toast will tie up a small portion of the screen for a limited period of time, and then time out.

Appendix A: Devices

Device	**Forerunner 920**	**Epix**	**vivoactíve**	**Fēnix 3**
Resolution	205x148	205x148	205x148	218x218
Colors	16	64	64	16
Buttons	Up Down Enter Back Menu	Up Down Enter Back Menu	Back Menu	Up Down Enter Back Menu
Touch Screen	No	Yes	Yes	No
Text Font	Digi	Roboto	Roboto	Roboto