DEVELOPER GUIDE

Pebble UX Design

Chapter 1

Pebble UX Design

Learn the fundamentals of UX design for Pebble

Section 1

Introduction

CONTENTS

If you've designed and created apps for Android and iOS, you'll no doubt be familiar with many of the most common design principles and paradigms for each platform.

The Apple iOS Human Interface Guidelines places strong emphasis on UI designs for iOS 7 that are simple, clear and deep, meaning that the UI won't compete with app content, text is always legible, and visual layers serve to heighten the delight of iPhone users.

Likewise, Android emphasizes Principles of UI design that are intended to keep the best interest of people in mind. Toward that end, Android recommends that when creating the UI for your app, you follow these basic principles: **Enchant Me, Simplify My Life, and Make Me Amazing**.

You may wish to review these guidelines for iOS and Android, if you haven't already, before you begin designing the UI for your Pebble watchface or watchapp.

The Pebble Platform

Pebble connects to iOS and Android devices using Bluetooth, alerting users with a silent vibration to incoming calls, emails and messages.

Pebblers enjoy the convenience and power of accessing information with a simple glance or flick of the wrist.

Navigating with simple button clicks, Pebblers can perform a wide range of tasks, including notifications, remote control, monitoring sports and fitness activities, and more.

With iControl, Pebblers can remotely manage and control the alarms and security systems of their homes, even toggling their lights on and off.

Using the GoPro camera, record and take pictures from with the Pebble app for

Pebblers can their Pebbles, GoPro.

Tiny moments of awesome!

Tiny moments of awesome is one way to think about the experience and power of the platform, if you look at the iControl, GoPro and Yelp Pebble apps.

Find choice restaurants in locations with a flick of the wrist, with the Pebble app for Yelp.

The Pebble Ecosystem

Pebble lets you install from a wide selection of custom watchfaces, and then lets you change and remove them as you wish.

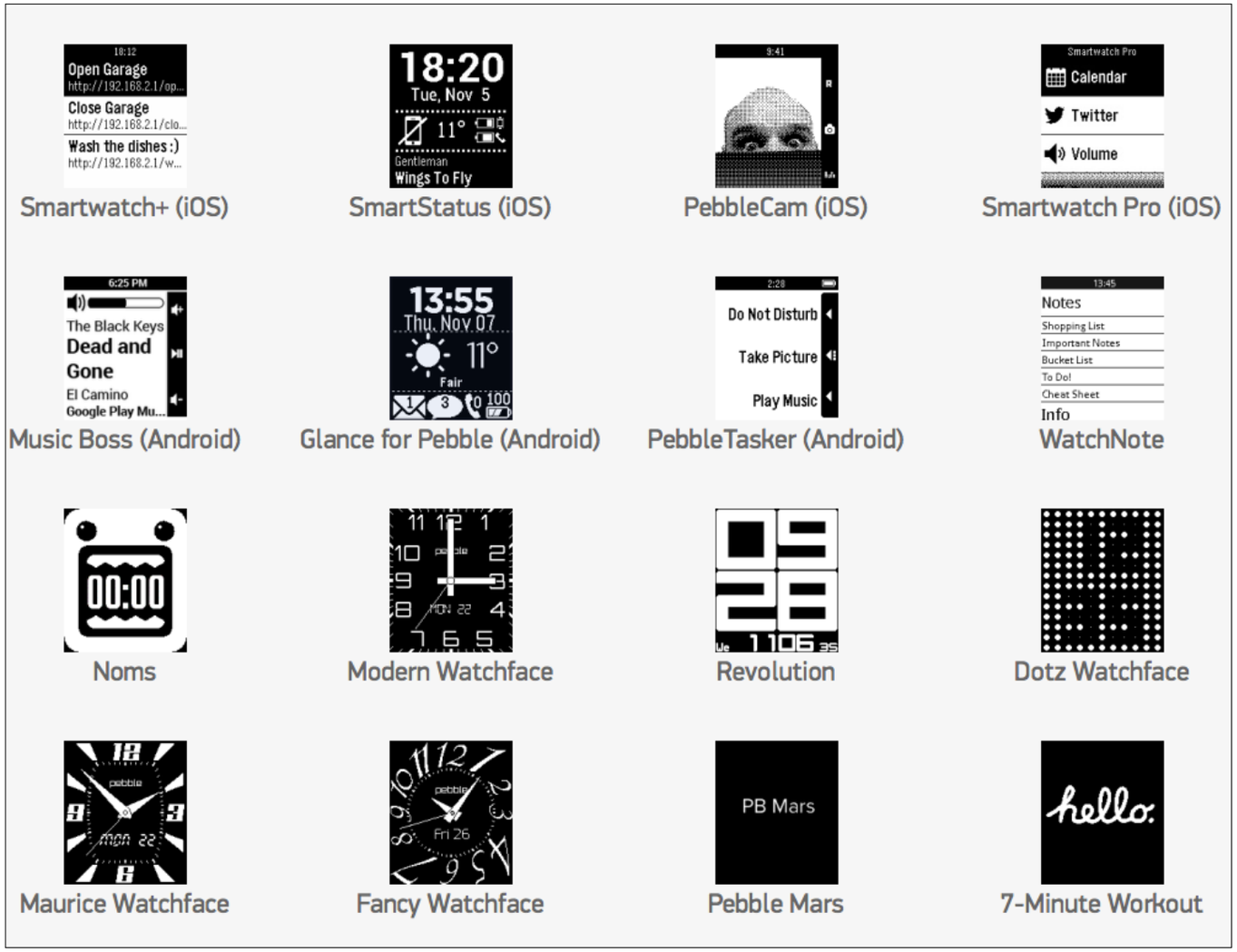
People typically interact with one app at a time, either a watchface or a watchapp, which are the two primary types of apps available for Pebble users.

Beyond this basic functionality, the beauty of Pebble is that people can expand the power and capability of their smartwatch by choosing to acquire, download, and install a host of Pebble apps.

The types and categories of these apps range from notifications and fitness to tools and games, and whatever Pebble developers can create and imagine..

A wide range of simple and elegantly designed watchfaces and watchapps are available for Pebblers to download and install on their Pebbles.

The number of available watchface and watchapps is growing by leaps and bounds, offering both enhanced functionality and new opportunities for Pebble developers who are creating next generation watchapps and wearable software.



The Pebble appstore

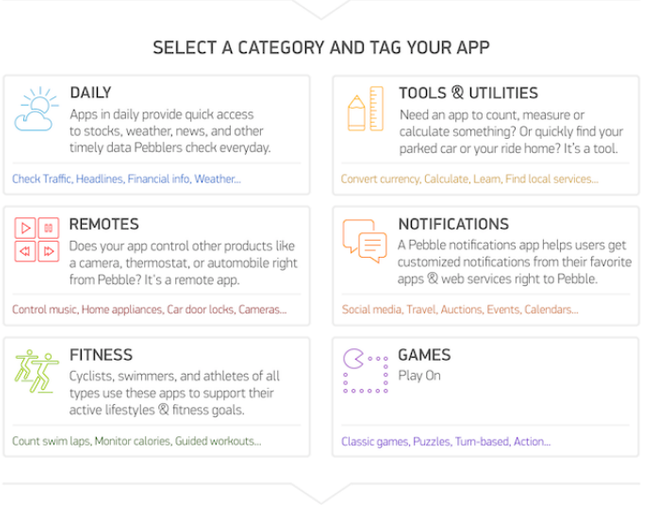
Pebble developers have long been asking for an online app store that lets them publish, promote and distribute their watchface and watchapps worldwide. That's now coming in early 2014 and will go live with the forthcoming release of Pebble SDK 2.

Pebble appstore will be included in the official Pebble smartphone application. It's the first place where Pebblers will look for the latest and greatest Pebble apps in release.

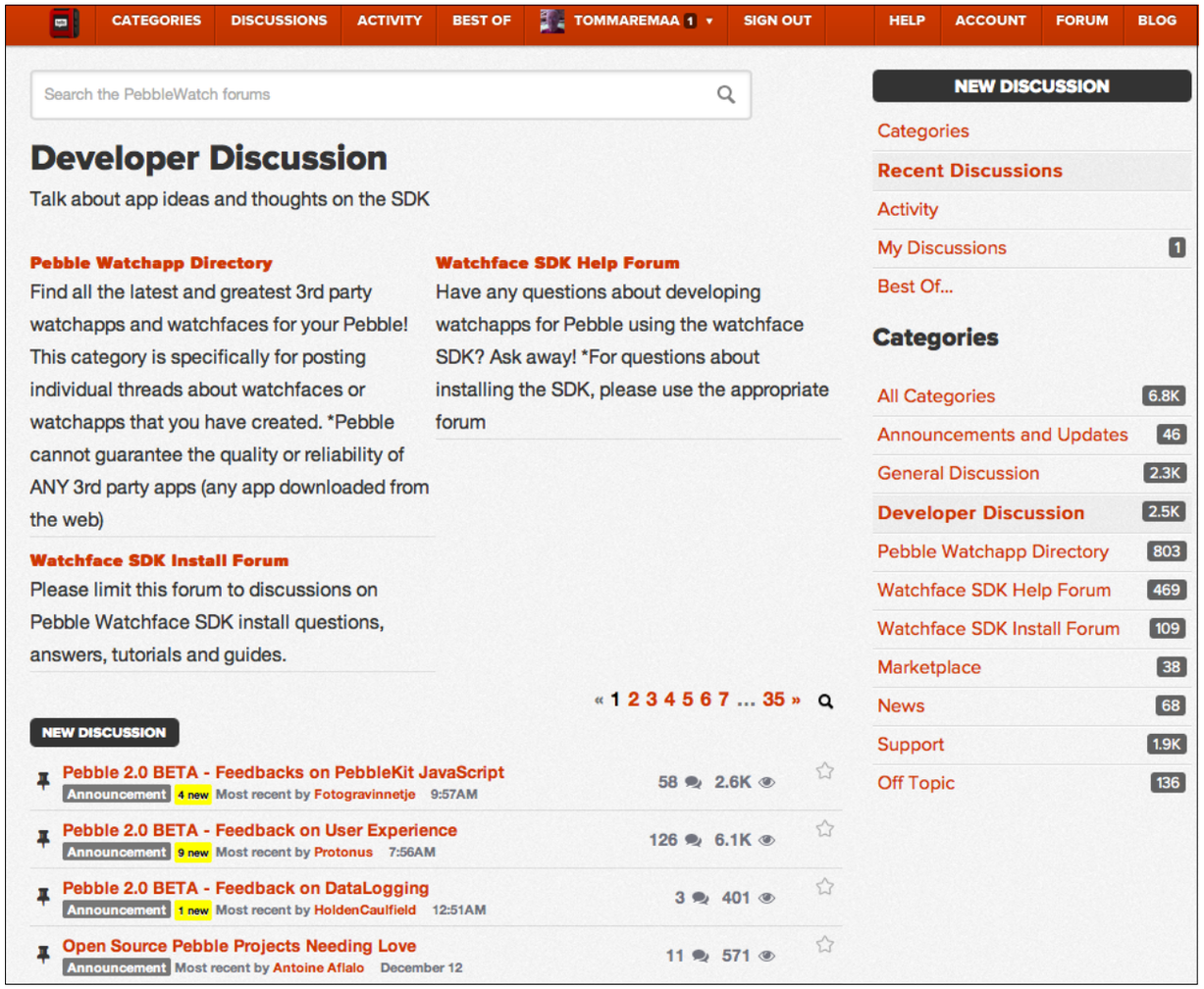
As for developers, the Pebble appstore will become the best and surest way to promote and distribute their awesome Pebble apps to users.

The Pebble appstore will include watchfaces and six different app categories for Daily, Remotes, Sports & Fitness, Notifications, Tools & Utilities and Games.

In each category, Pebble will feature a selection of the best as well as new apps that are available for the platform.



The Pebble SDK



Pebble provides third-party application developers with a robust and evolving API for creating, building and deploying great Pebble watchface and watchapps.

The API includes the latest Pebble software builds, libraries, tools, documentation and sample code.

Pebble has also actively engaged the developer community with forums for discussion and support in multiple categories. The community of Pebble developers continues to grow worldwide, as Pebble creators share and exchange valuable information.

Classes of Pebble apps

Beyond watchfaces that simply tell time in useful and creative ways, you'll find a new emerging class of Pebble apps that provide Pebblers with greater control over their mobile devices, pushing out valuable data for a range of personal and fitness activities.

Pebble app developers have stepped up to produce apps that are aimed for specific use cases, like geo location and travel information. remote control of home security, and monitoring of personal activities. In Chapter 3, you'll discover a small sampling of these apps from third-party developers who are dedicated to the Pebble platform. The enhanced functionality provided by these apps have served to extend the power of the Pebble platform. Each app brings its own unique design to Pebble.

Pebble developers can also leverage the PebbleKit native libraries to add Pebble support in iOS or Android applications. Beginning with the release of Pebble 2.0, apps can also use PebbleKit JavaScript to connect to the Internet, access the phone GPS and store information. PebbleKit JavaScript works perfectly on iOS and Android with only one source code.

iOS and Android apps can use the **Sports API** to control the Sport and Golf display on Pebble.

RunKeeper and Golf were among the very first mobile apps to add support for this API.

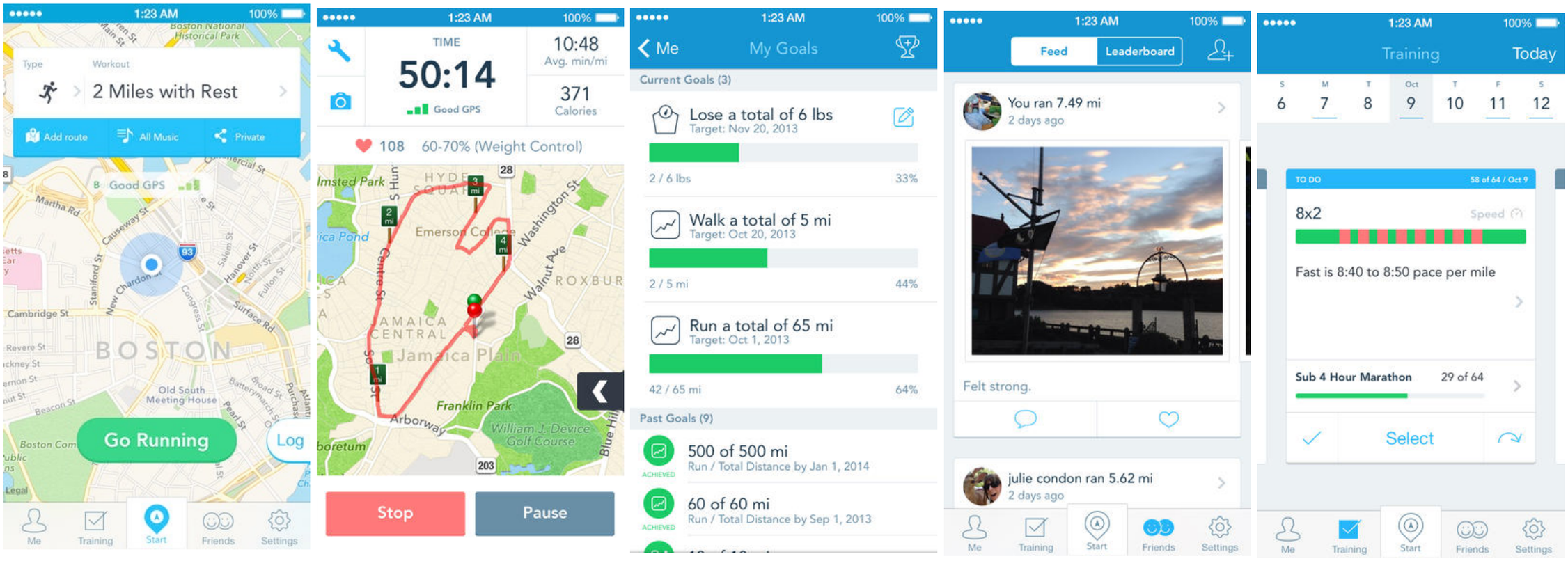
The RunKeeper sports watchapp (above left), showing time, distance and pacing for a Pebbler who is running or working out. The Pebble Golf watchapp (above right), displaying par, hole and distance to the hole. Both are built into every Pebble.

Power of the Platform

RunKeeper enables you, in effect, to turn your mobile device into a personal fitness trainer. You can track your running, walking, cycling, hiking and other physical activities, using GPS on your iOS and Android mobile devices.

RunKeeper and Golf paved the way for the next generation of Pebble apps, like GoPro, iControl, Yelp and Foursquare.

The design of these apps leverages the talents of UX designers working closely with Pebble programmers and coders.



Both designers and programmers can unleash their creativity when building next generation Pebble apps.

Chapter 2

UX Design Basics

Designing a great UI for Pebble takes time, effort and creativity. You'll need to understand platform conventions and design patterns, and you may have to iterate and test your design through multiple revisions. In the end, though, the hard work will pay off. Users coming to the Pebble platform, either from Android or iOS, will enjoy and delight in the apps you've created that enhance their channels of communication, productivity, life and play experiences.

Basics

The Visual Canvas

The Pebble platform is, at once, unique and challenging, given that you'll be designing and creating apps on a smaller visual canvas than a mobile device. The display is black and white, bitmapped, with a screen resolution of 144 x 168 px. Time can be configured to constantly display in a range of fonts and sizes. People navigate by means of button clicks, windows, and menu items.

By default, every Pebble comes with controls for enabling music, alarms, and watchfaces. The arrows (below) indicate middle and back buttons to navigate from one app to another.

User control of apps is only a single click away..

.

Watchfaces

Pebble provides users with a **carousel** of watchfaces, ranging from Classic Analog to Simplicity. And of course any app that the user wants to install.

Watchfaces are at the center of the Pebble user experience. With a simple click of a top or down button, Pebblers can navigate through a variety of

watchfaces displayed on their Pebbles. Depending on mood or context, Pebblers can easily change watchfaces.

Designed to run for long periods of time, watchfaces provide people with updated and timely information at a glance. A watchface may contain a single screen that prominently displays the time and optionally some other bits of information, like calendar dates or even weather conditions.

A watchface typically has these attributes :

- Displays at full screen without the system status bar
- Updates only once per minute or second (in most cases)

- May be frugal in its use and handling of system resources, like power and energy consumption, and display updates to preserve valuable battery life
- Can be a default app automatically displayed when the launcher menu times out
- Can't interact with the user, except through gestures and the accelerometer. No buttons!

Watchapps

Watchapps are launched through the Pebble system menu, providing users with greater functionality and user interaction than watchfaces. The Pebble user experience is notably enhanced, as you navigate content through multiple screens and menus, clicking one, two, three or all four of Pebble's built-in buttons.

Watchapps provide users with a powerful, yet easy to use device for accessing web or other informational services, tracking fitness and sporting experiences, controlling remote devices, playing music and much more.

For Pebble watchapp developers, the possibilities are only limited by one's imagination.

A standard Pebble watchapp:

- Uses the system status bar at the top of the display showing the time, although apps can also be made fullscreen, removing the status bar.
- Updates the display more frequently than a watchface
- Responds to user interaction and button presses

Designing for Pebble

Pebble apps communicate a user experience that is rich in notifications, interaction and control of events and devices that matter in one's mobile and digital life.

This is achieved through the use of **Flat UI Design** for most Pebble watchface and watchapps. Flat design embraces these elements:

* **Absence of depth, that is, visual flatness for ease of navigation and deference to app content.**
* **Simple yet cool UI components, that is, icons, scrollers, windows with useful information for Pebblers.**
* **Smart use of typography for text and number that are instantly viewable with a simple glance at the display.**
* **Imaginative use of black and white (monochrome, no color) displays.**
* **Minimalism in design that lets you perform a wide range tasks, from sports and fitness to remote control of devices.**

The **Pebble Card** app serves as a good example of flat UI design at work. Fonts for date and time are set in the most readable size, according to the value of the information required by the user at a glance. The sharp black and white display of that information enables easy access to the data on city location, weather conditions (Light Rain) and temperature.

Pebble Card example is discussed in Chapter 4 of the UX Guide.

For more information about the principles of Flat UI Design, check out these websites:

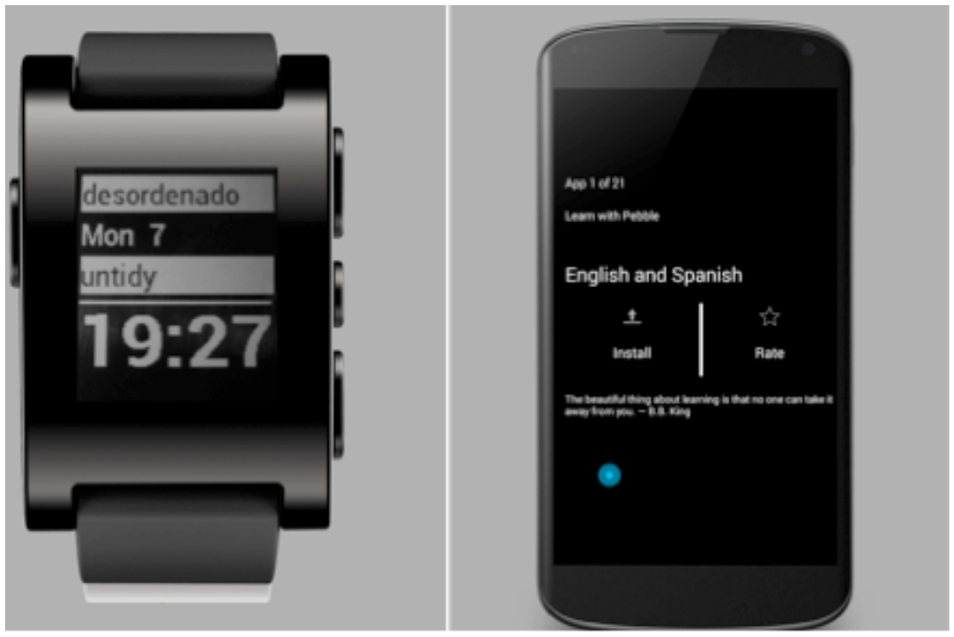
- Sitepoint with a getting started guide
- Pinterest for Flat UI design examples

Instantly Engaging Pebblers

Understanding how people expect to interact with apps on Pebble is essential for creating good and intuitive user interfaces.

Pebblers will enjoy that **moment of delight** that comes from a simple glance at a watchface or watchapp that displays exactly the right bits of information at the right time.

The **Pebble Watch Tutor App** is one example of a Pebble watchapp that immediately engages people with a simple glance at the display.



Each time you glance, you learn a new word and build your foreign language vocabulary, a kind of mini-language tutor. The Android version of the app lets you select watchfaces with different languages. The app is set so that with each passing minute, a new word pops on to your Pebble.

A mini-language tutor watchapp that displays a new word each minute

Check out the Pebble Watch Tutor App, if you're curious.

Patterns of Interaction

Pebblers typically interact to information displayed on their smartwatches in a variety of ways. Some of those ways are described here.

Being notified

Notifications draw user attention by means of direct display of information or by means of Pebble vibrations.

Glancing at text and/or numbers (digits)

Pebblers will glance at their displays to get concise, up-to-date information, like text or numbers. For this type of integration, uncluttered screens with wide and readable fonts are ideal.

Active reading text and/or numbers

If your watchapp user is interacting with Pebble, reading text actively and carefully, you'll need to choose a font that is appropriate for this level of interaction. In most cases, you'll find text that displays at a smaller font size may be appropriate.

Typography Sidebar

Whether Pebblers glance at text or actively read the text and/or numbers displayed, you'll need to choose fonts that are ideal for readability.

Of course, the experience of reading text on Pebble will vary from one user to another. The context in which your watchapp is used plays an important role in deciding which system fonts to use, or if you should create your own custom font.

Any choice of font involves a compromise between readability, density of information that is displayed, and pure design aesthetics.

The Pebble System Font Set

The Pebble OS supports font sizes up to 47 points. If you need a larger font size for your app, you can create your own.

Depending on the app, Pebble developers have ingeniously employed a variety of font sizes, styles and typefaces when displaying text and numbers on Pebble. In each case, Pebblers have enjoyed the benefits of this effort to convey information in creative ways,

Pebble recommends that you use the default system font, that is, Raster Gothic Condensed, for your watchface and watchapps.

For more information about fonts

Refer to Handling Pebble Font Resources in the Pebble Developer Guide, which describes the options available for choosing fonts, either using the system font or creating a custom font.

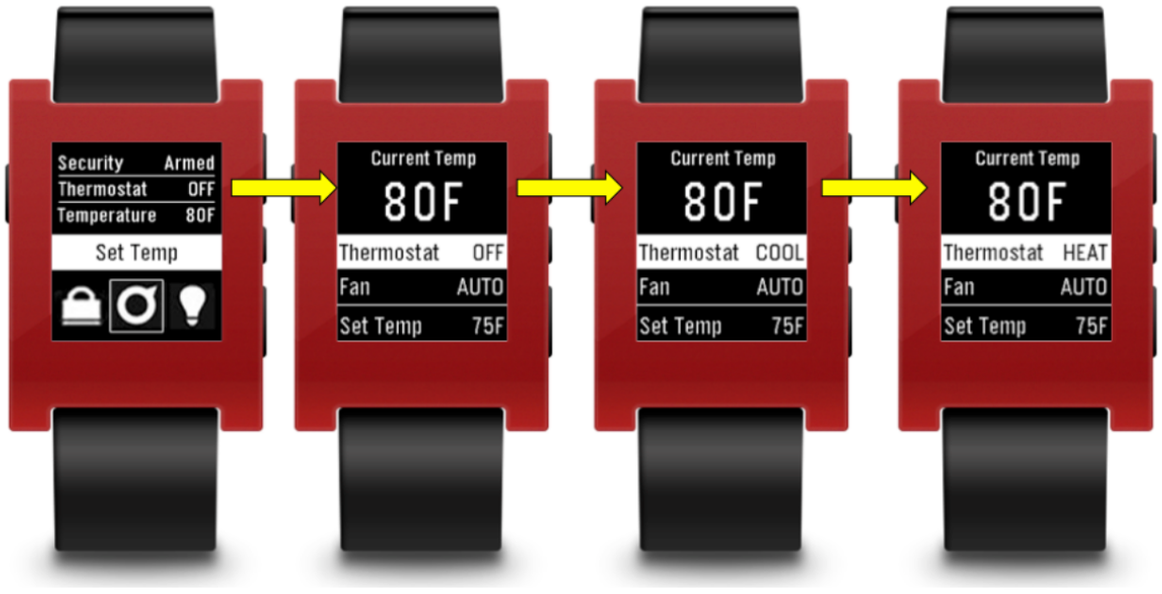
Tip: If your watchapp user needs to read text or numbers while glancing at the watch, **Pebble recommends that you use at least a 28 pt font size and a bold style.** The Text Watch app, for example, uses a bold font sized at 42 pt for the hours.

Pebble uses the bold system font at point size 24 for the titles in system menus and as the default or large notification font. For even larger bodies of text,

Navigation

Navigation from window to window in a Pebble watchapp should be as intuitive as possible for Pebblers. Think simplicity and ease of use.

Your watchapp should simplify navigation from window to window, button



click to button click, depending on the type of content and visual information you want to convey.

In the above example, the **Pebble iControl watchapp** lets people remotely control various temperature settings for their house. A simple button click takes them to the Set Temp setting screen. From there, you can check the current temperature, click the Thermostat OFF, COOL or HEAT.

The **iControl** app lets Pebblers change their settings quickly with one or two button clicks. The display shows all the menu options for various temperature settings clearly, enabling easy control of settings and navigation from one window to the next, with a few button presses.

Button clicks and tap events

Pebblers navigate the landscape of watchfaces and watchapp by means of button clicks and tap events. These are some user interaction guidelines you may wish to consider:

* Ensure your app responds quickly to notifications and events.

* Ensure, too, that information is displayed without delay on the screen.

* Scrolling via button clicks in menu items also occurs without delay.

* Tap events can create a new form of interaction with watchfaces or watchapps. By enabling tap events you can quickly transition between windows or offer new bits of information. Tapping also can take advantage of Pebble's accelerometer capability.

Tip: Vibrate is a great way to notify the user of a special or significant event.

Types of clicks

Pebble provides support four different types of button clicks:

* **Single-click**. Detects a single click, that is, a button down event followed by a button up event. It also offers hold-to-repeat functionality.

* **Multi-click**. Detects double-clicking, triple-clicking and other arbitrary click counts. It can fire its event handler, which triggers an event, like a change of screen display, on all of the matched clicks, or just the last.

* **Long-click**. Detects long clicks, that is, press-and-hold for a specific interval, which a Pebble programmer can configure in code.

* **Raw**. Simply forwards the button events.

In your Pebble app, you can set up click handlers programmatically in each application window in order to process button input and call the necessary event handlers in code.

The techniques for accomplishing this are described in the section Building Great User Interfaces in the Pebble Developer Guide. The section is essential reading for programmers working with UX designers.

A few bits about the Pebble UI framework

For programmers creating the UI for Pebble watchface or watchapps, it's important to understand how the various design components of the Pebble UI framework work. You'll find this information, described in detail, in the section Building Great User Interfaces in the Pebble Developer Guide.

In brief, the Pebble UI framework is comprised of the following components, which you use in designing and creating your Pebble apps:

• **Event handlers**. A set of event handlers that you use to interact with various components.

• **Windows**. The display of Pebble watchapp views and activities. Apps can have multiple windows and enable user interaction. A window can also handle various click inputs, like short and long clicks, hold-to-repeat clicks and double clicks.

• **Click handlers**. A event handler for a window that captures button events.

• **Layers**. Visual objects displayed in windows on the watchapp screen.

• **Graphics routines**. Used to draw pixels to a graphics context provided by a layer.

• **Animations**. A base component used to create arbitrary animations, like simulating a linear motion with non-linear timing.

Each Pebble UI component is unique yet designed to interact with other components in the Pebble stack.

Designing icons

Designing icons for Pebble apps offers a unique challenge. Icons need to be bitmapped, and communicate information easily at a glance. All app icons are flat and free-standing, that is, without a background shape behind them, like the rounded rectangles in iOS. Menu icons should be monochrome 28 x 28px in PNG image format, without transparency.

In the Pebble Yelp watchapp, the author adapted the iconography from Yelp's website, creating equivalent icons for Pebble. The icons were bitmapped, 26 x 26 px, as well as smaller, 16 x 16, depending on navigational requirements for the user.

Most any graphic program will enable you to create these icons at the specified pixel size.

Once you decide on the visual look you need, you'll have to build that icon for your app, pixel by pixel.

Tips to optimize your design

There are a number of techniques you can use to optimize the design of your Pebble watchface or watchapp. The following list is by no means all-inclusive, but should give you an idea of design choices that you may wish to consider:

* Keep your app focus on the most frequently-used watchapp scenarios

* Defer complex, secondary or lengthy tasks to a phone or PC app

* Optimize your app for “perceived” speed

* Minimize the number of user clicks (up to and including none)

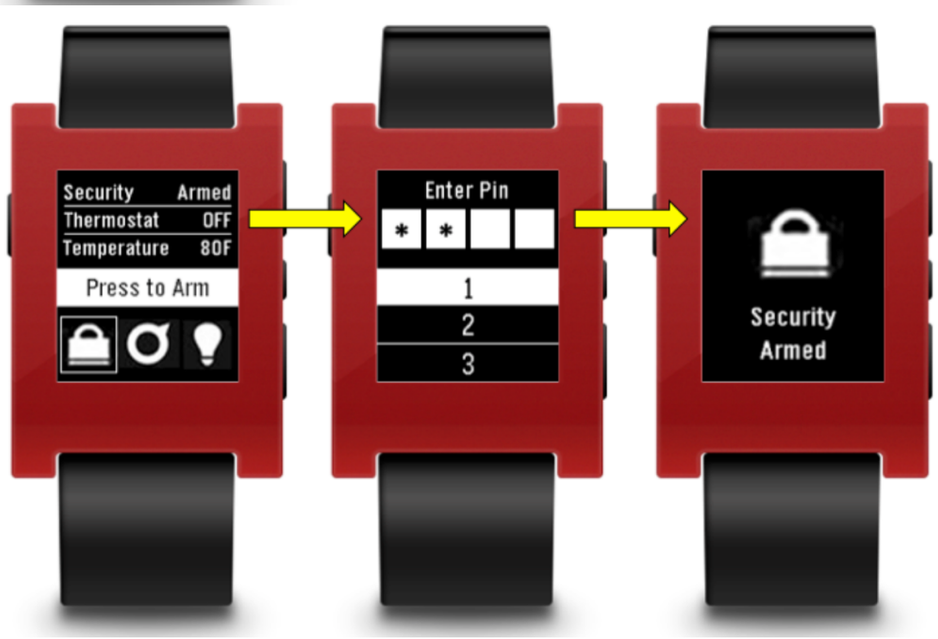
* Use animations to “mask” a lengthy process

* Provide shortcuts, where possible, for power users

* Avoid intrusive offline error messages

* Avoid delays when the user starts your watchapp

* Show a demo of the app's features when it's properly configured and be sure to provide an easy path to the setup process for the user



Integrate smartly with mobile, PC and web apps

Imagine if you're able to connect any web service to your watchapp or watchface. This means you can integrate with mobile, PC or web apps and create new and creative designs that push the envelope of what is possible.

Web service integration is an important facet of the Pebble user experience. Some principles you may wish to consider:

* **If your app requires a user setup, ensure that you guide the user through the process, logically, step by step, from device to device.** You may need to test and re-test the setup process, eliminating user confusion or frustration.

* **Make the Pebble watch as valuable as possible when it's disconnected or the configuration is missing.** For example, when designing a Notes app, consider storing the notes on the watch, so they can be displayed when there is no connection.

* **Start your watchapp immediately.** You accomplish this programmatically by caching the latest data for offline use. For more information, refer to the section dealing with Persisting Data in the **Pebble Developer Guide**.

An awesome music app for Pebble

MusicBoss is a good example of flat UI design, with enhanced functionality for Pebblers.

Note the clean lines, smart use of fonts, sliders for volume control, icons that let you control music playback, with the time shown in the status bar at the top.

You can perform a variety of tasks with simple button clicks on Pebble, like adjusting the volume with your Pebble while you're listening to music, launching your current Music App from your Pebble, switching music/audio app using the Music Boss Quick Access App List.

You can use the existing Pebble Music Watch App or Music Boss Watch App. The app is also integrated with the Glance for Pebble watchapp, and integrated with Tasker: http://musicboss.ca/tasker.

The Volume Mode (with existing Pebble Music Watch App) allows you to quickly and easily adjust Volume.

Flat UI Design at its best

An example of Flat UI Design for playing music on Pebble with volume slider, simple yet clear text elements, and information, like time, for Pebblers at a simple glance on the wrist.

Chapter 3

UX Design Patterns

Understanding UX design patterns is essential for developing great Pebble apps.

This chapter explores some of the basic design patterns that you need to consider when creating the UX for your watchface or watchapp.

UI Design Patterns

CONTENTS

Pebble Design Principles

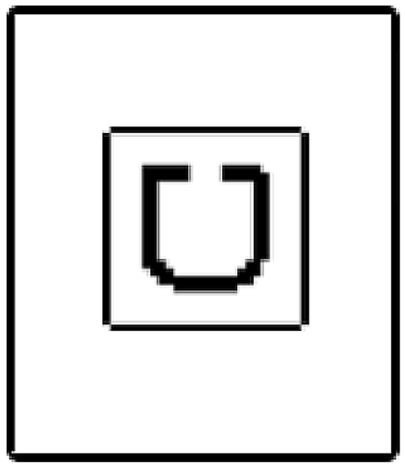
- Think of your app's core functionality and design around that
- Focus on the simplest way for the user to interact with a complex data set
- Keep the pattern of engagement with the user consistent and intuitive
- Create a single feature that distinguishes your app from other apps
- Design a killer use case for your app
- Readability is really, really important

- Identify the one feature that's going to amaze and excite the user and then build around that
- Don't model your app around the Pebble UI. Create your own UI.
- Design around speed of interaction, not Pebble's UI for navigation

Design Patterns

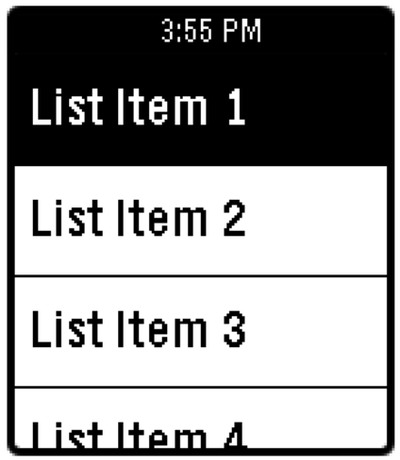
Undestanding Pebble design patterns are critical in creating, building and publishing great Pebble apps.

The following sections describes some of these patterns, which you wish to consider when designing your UX.



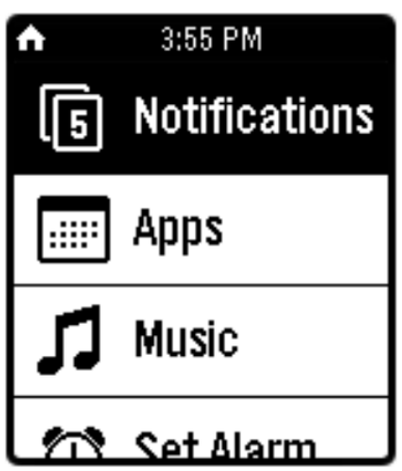
Splash Screen

A splash screen on launching your app exposes your brand and also gives the rest of the app a few seconds to load.



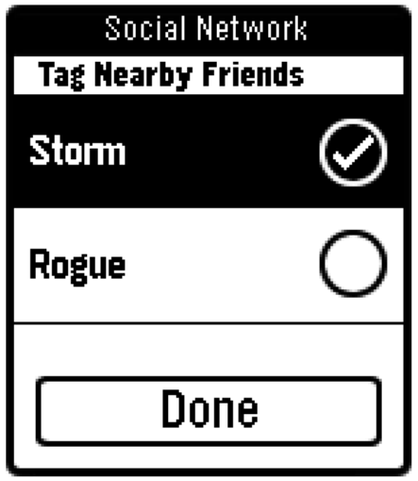
Simple List Menu

A simple list menu is the most basic way to display an expandable number of items from which to choose.



List Menu with Icons

Icons can help users quickly identify the different sections or options, particularly if the list is being used as your app's main menu. Subsequent screens can then carry smaller versions of the icons in the screen header to reinforce a user's sense of location.



Multi-select List

If your app lets a user select multiple items in a single view before performing an action, use the multi-select list view. Pressing the center button toggles a checkbox (or radio button) on and off. This pattern also contains a frame at the bottom with a selectable action button that can be eventually reached after pressing Down a certain number of times.

Mismatched List

Avoid using multiple patterns in the same list view.

List headings

Headings in list views are used to designate separation between different parts of a list. They are narrower than full-height list menu items. A black font on white space helps (as opposed to white font on a black heading bar) avoid confusion of where the user's selection focus lies.

Fat List

Fat lists support a sub-line that serves to bubble up high-value information without leaving the same list view. The bold/unbold font treatment serves to emphasize the information hierarchy.

List Menu Marquee

If a list item label doesn't fit within the width of Pebble's screen, marqueeing after a second upon focus is a nice way of exposing the remainder. This is most suitable for lists of dynamic content where the entire label is important to know (for example, Sports statistics).

Center Button Action

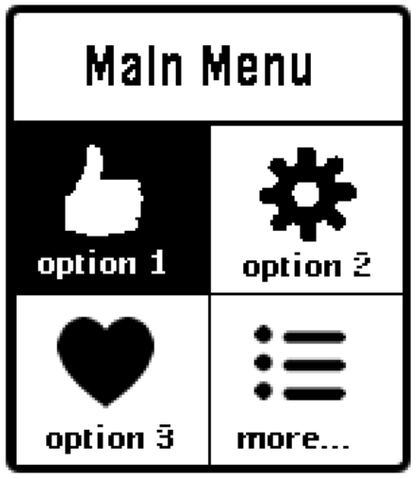
If your app has a primary function that you want to assume priority over everything else on the screen, map it to the center button. A well-designed icon in the right margin of the action bar is the key to making this screen clear to the user.

In this example, a flag icon is used to represent the beginning of a timed exercise.

Center Button Action Overload

When the primary action of your interface toggles between states or modes, changing the icon after the button is pressed communicates this without loading a new screen and changing context for the user.

Tapping on Center key, in this examples, toggles between Play and Pause states.

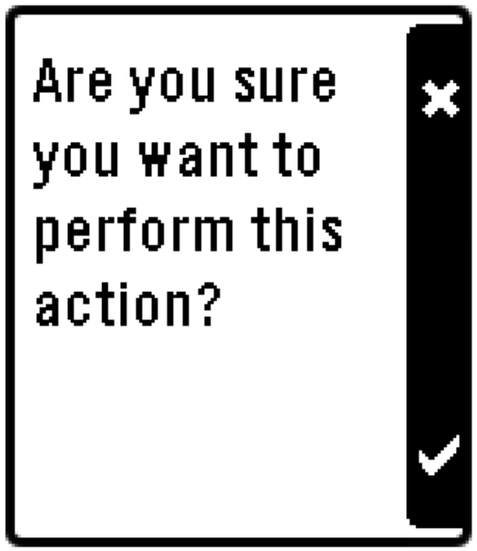


Grid for Larger Icons

A **2x2 grid** supports larger icons and a fatter heading.

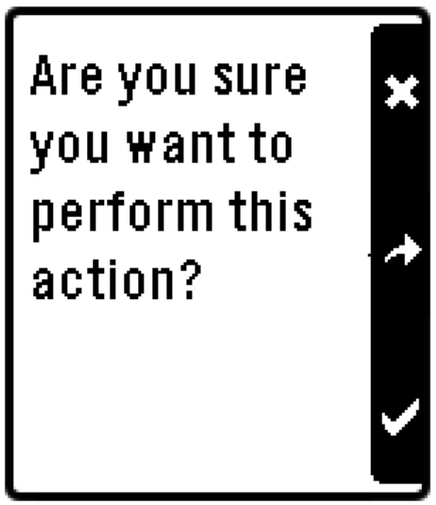
Use this for maximum impact to display a main menu of functions, especially if you have exactly four of them that are unlikely to change. More options can always be accessed from the fourth tile that expands into a list view.

Decision Screen with Action Bar



The **action bar** is the primary way of exposing Yes/No decisions to the user. It maps **Affirmative** actions to the bottom button on Pebble (Yes/Ok/Confirm), and **Dismissive** or negatory actions (Cancel/Dismiss/No) to the top.

Standard icons that suit this pattern well are checkmarks and [X]'s. The rest of the screen is left to contain the accompanying dialog.



Action Bar with a More menu

If there are other actions a user can take besides an affirmation or dismissal, you can map them to the center button on a decision screen. Typically, these may include **Remind me Later**, **Don't ask again**, or otherwise a set of less popular options that accompany these two primary ones.

Popup List also known as a More Menu

A **popup list** menu is best used to expose **More** options when a decision screen needs it.

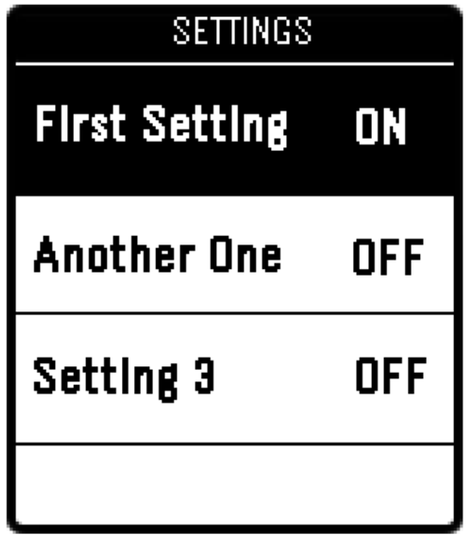
It can also be used to expose a **contextual menu** if a user long-presses a list menu item.

Full screen slides - Panel Dots

Panel dots are a common pattern on mobile devices and help to arrange and organize a series of full-screen items such as text messages, pictures in an album, or contacts in a group. Using this pattern will limit the number of items in a series to how many dots fit across the screen horizontally. Allowing them to bleed beyond the margins is not encouraged. In that case, use the unlimited design pattern.

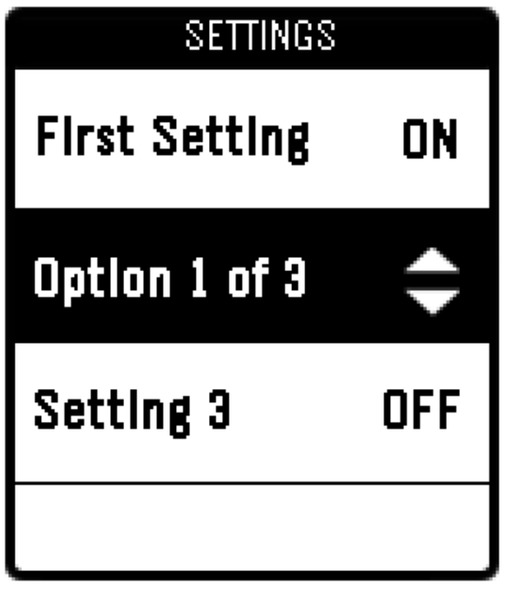
Full screen slides - Unlimited

This variation on the panel dots pattern supports an unlimited number of successive slides, making it more suitable for larger sets.



Setting - On/Off

Settings that map well to either an On or Off settings should use this pattern. The current setting state is displayed in the right margin. Avoid grouping the On or Off words along with the name of the setting itself (for example, Notifications OFF and Notifications ON as single strings).



Setting - Nested List

If a setting has multiple options that can be selected, using a nested list is an effective way of exposing them without leaving the current view.

Pressing the center button while focused on a nested list will map the top and bottom buttons on Pebble to scroll the nested list.

Pressing the center button again leaves this mode and allows the top and bottom buttons to once again scroll the master list.

Chapter 4

Pebble UX Design Examples

From a carousel of watchfaces at startup to thousands of Pebble apps to choose from, Pebblers are enjoying the ever-growing number of apps available from Pebble developers worldwide.

These are just a few examples of the great apps coming to the platform.

They are intended to illustrate certain design principles and techniques that may be useful as learning tools for UX designers.

A Pebble Apps Sampler

Watchfaces

Beam UP excels in the features associated with minimal watchface design, combined with animation. The app relies on small, compact animations that display 15, 30 and 45 seconds past the minute.

On the minute change, the digits are beamed up, then animated.

This use of animation, with digits in a readable font at the center of the display, instantly catches the eye of Pebble users when they glance at their wrist to get the time.

Check out Beam UP at the play Google store where you can get this and other watchfaces with the developer's free Selector app.

SmartModern

The design of this watchapp is based on zalew's exquisite Modern watchface, displaying weather, date, and iPhone battery status on the main watch face. The watchface is robust in the information that it displays, showing the next two calendar appointments; alerts (with vibration) for

calendar appointments. The display also shows the currently playing song. There are graphical indicators while updating and for lost phone connections.

SmartModern is the analog version of SmartStatus. Both apps require the App Store version of the Smartwatch+ iOS app to work. Check out Smartwatch+, if you're curious and want to explore its features.

Marked could not be simpler yet more powerful its minimalist design. The watchface includes date, time and day.

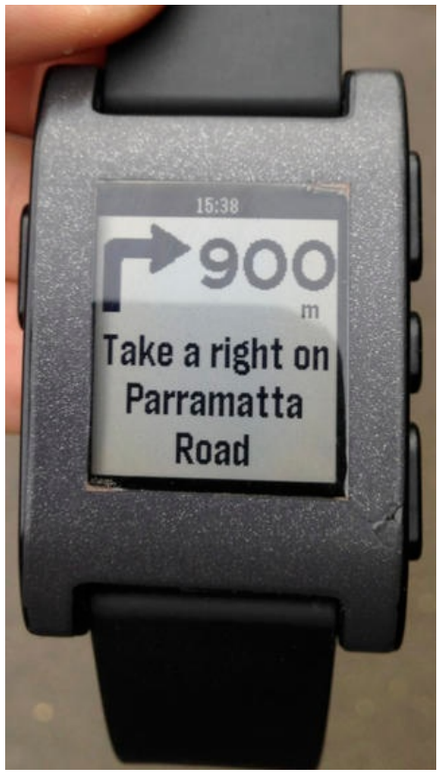
The current day is marked with a minimal underscore.

Note the use of Futura fonts in the design. Future is clean and readable for Pebblers glancing for this valuable on their Pebbles.

Check out Marked if you wish to download and install this cool Pebble watchface.

Watchapps

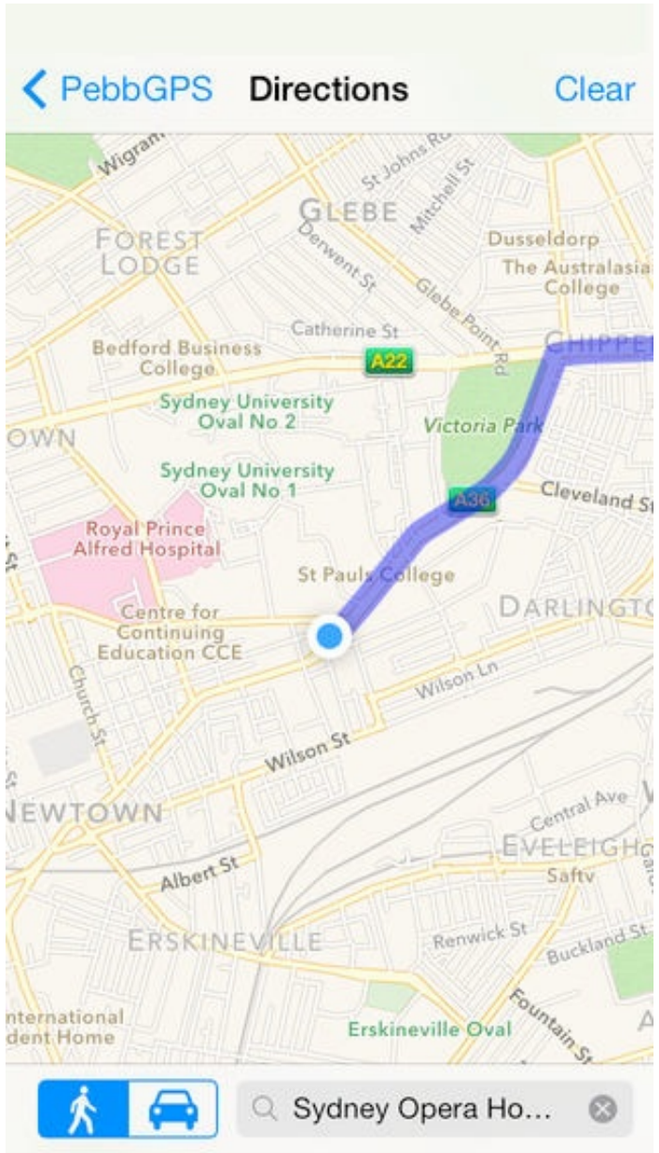
PebbGPS lets you to put a mini-map on your Pebble and get turn-by-turn directions to a destination. The design is clean and the maps are readable as you navigate to your destination.



You install the PebbGPS watchface from within the app.

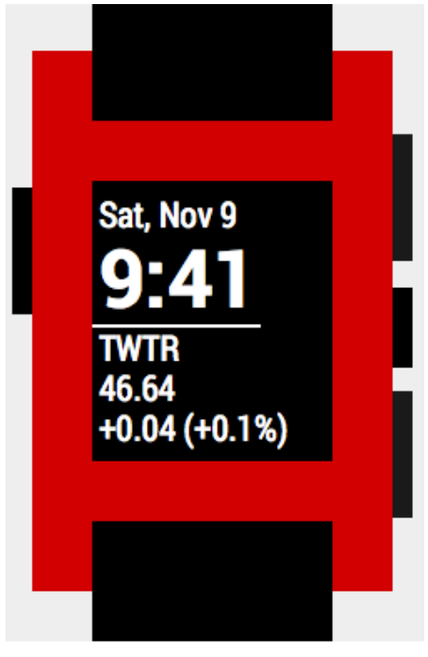
Both driving and walking directions are supported in the latest version.

The turn-by-turn directions appear on your Pebble, as either text, arrows for the next turn or on the map. Distances to the next turn point are in metric and imperial.



Check out PebbGPS and see how maps are implemented in this cool Pebble app.

Pebble Cards from developer Keanu Lee uses the metaphor of a collection of cards to display a wide range of useful information on your Pebble.



You can customize the cards, too.

You add cards that contain headlines from your favorite RSS feed, your next Google calendar appointment, local weather, stocks, a world clock, and even a simple note. Customize your cards from within the Pebble App on your smartphone.

The date, time, and card are always visible on the Pebble. You use the up and down buttons to cycle through all your cards. You press the middle button to refresh the current card. You hold the middle button to see more details.

From a design perspective, Pebble Cards is **an app that displays short bursts of information on cards, instantly captured and absorbed by Pebblers.**

In design, multiple customizable cards can make up a deck that can be scrolled through, updated, or expanded to expose more details about the information pertaining to a particular card.

MazeCrawler is a 3D, first-person, maze-navigation game for Pebble, which is compatible with SDK 2.0 firmware. It's the foundation for the forthcoming dungeon crawler PebbleQuest.

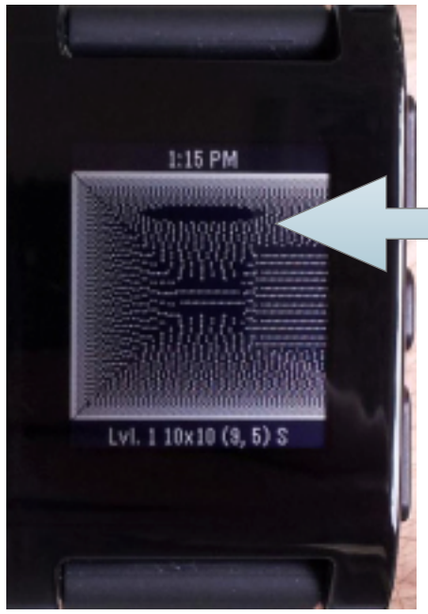
Games designs are often difficult to implement, but MazeCrawler works successfully on multiple levels. The goal of the game is to complete each maze as quickly as possible, so you can accumulate more points.

Mazes in the game, according to the developer, are procedurally-generated with an entrance marked by a hole in the ceiling and the exit marked by a hole in the ground.

For navigation and movement through the maze, you use various controls on your Pebble.

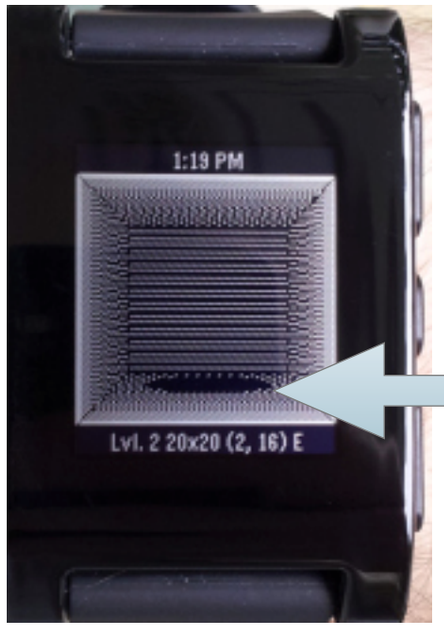
You use the **Up button**, pressing once (or hold) to step forward, or pressing twice to turn left.

The **Down button** is used to step back. You press twice to turn right. The **Select button** opens the in-game menu and makes selections within menus. The **Back button** closes the in-game menu, returns the player from gameplay to the main menu, and while in the main menu, exits the game.



The Hole in the Ceiling

If you look closely, you'll see a hole in the ceiling. That's how you wound up in the tunnels of MazeCrawler. Now you must find a way out. . .

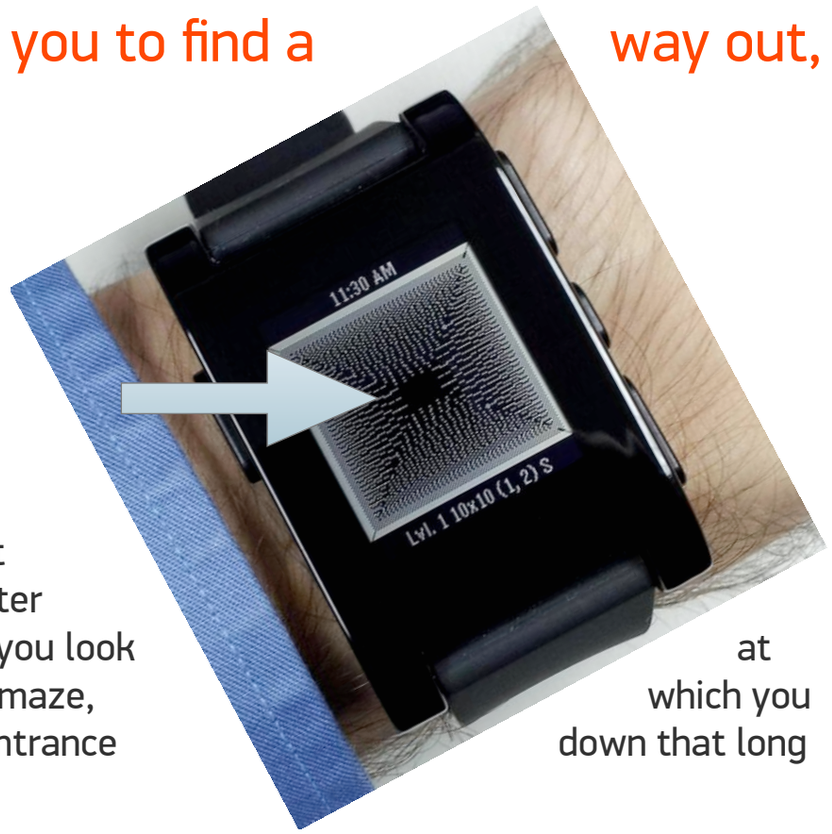


The Goal

Your goal lies before you! That hole in the ground is an exit . . well, maybe. . .

Be careful, it only seems to lead down through the dark corridors of yet another maze...

MazeCrawler illustrates the potential for developing cool games on Pebble, where navigation via Up, Down, Select and Back button clicks takes you through the labyrinthine tunnels of the maze, challenging you to find a way out, if you can.



The design is immediately engaging, even hypnotic in that it entices you to enter another world, if you look at the center of the maze, which you spot a possible entrance down that long hallway.