Proposed Wrist-Mounted Smartdevice

Juan S. Carrillo

November 27, 2014

1 Introduction

Wearables are growing in popularity. Recently, five out of the top ten crowdfunded campaigns on Kickstarter have been wearable devices. However, unlike the smartphone, it seems wearables have become more highly specialized, niche-specific products and fail to become standalone products in their own right. A large number of wrist-worn devices specialize as fitness trackers and do nothing more. More multipurpose products like Google Glass aren't completely independent; Google claims that in order for a "great on-the-go Glass experience, it's essential to pair Glass to your phone or tablet." Overall, there are very few wearables that don't need to tether to a smartphone or other device in order to have full functionality. As a result it's very difficult for wearables to displace smartphones, which have much more functionality and just as much mobility as typical wearables. In this paper, I propose a device that combines the functionality of a smartphone with the mobility and wearability as a smartwatch.

2 Background

The functionality of wearables are greatly limited by their size and form. For example, head-mounted displays lack a graphical, direct manipulation oriented interface like touch screen smartphones. Tasks as simple as sending a text message on Google Glass is significantly more difficult than on a smartphone because a user would have to resort to voice recognition software, which is highly error prone and requires the user to be in an environment with very little background noise. Smartwatches have very small screens which limit the potential of incorporating touch screens into a smartphone. Some smartwatches use buttons to navigate their menus, but unfortunately this method also is unsatisfactory because it takes several button presses to navigate to specific tasks. The small size of wearables also limits their processing power as well as battery life.

My design aims specifically to deal with issues and complaints common to smartwatches. The most important issue that I aim to address the "fat finger problem". The fat finger problem is that smartwatches have relatively small screens compared to the size of our fingers. Several approaches such as iterative zooming, buttons, gestures, skin buttons, knobs and movable watch heads in order to compensate for the small screen. However, I find these approaches unsatisfactory because they don't solve the fundamental problem of lacking space. Iterative zooming requires more taps and gestures to type in characters than a larger screen such as a cell phone or a tablet. Skin buttons make use of sensors that detect when a user a taps an icon illuminated on the users skin. Lighting conditions affect the effectivity of the sensors, and wearables should be equally effective regardless of the users setting. Buttons such as those included by the Pebble make it difficult for users to navigate a menu based system. Movable watch heads present moving parts which are often undesirable as they get worn over time.

Of these approaches, the only one that attempts to increase the size of the interface is the skin buttons interface. I propose using a larger device with a larger, multitouch touchscreen. This solves the basic problems associated with tiny screens. My design borrows heavily from the design of smartphones, and I hope that the familiar design will appeal to users and allow usage of my device more intuitive and easy. Ultimately, the main advantage of a larger screen is that navigating the devices interface is much more easier. A larger device also allows for longer battery life and increased computational power. Admittedly, a larger device is also more obtrusive and more uncomfortable for users; I address these issues in another section of my paper.

3 Physical Design

I propose a wearable device that is worn on the left forearm. It would be roughly five to six inches in length and would wrap around the forearm entirely. It would have two rigid, curved, durable sapphire multitouch touchscreen that would curve around the top and lower side of the forearm as in the diagram below, held together by a piece of hinged piece of material that locks into place on the other end of the screen section. This would allow the user to put on the device by sliding the device on to the inner forearm and then securing the device by locking it in place.

The device has a single camera beneath the user's palm that acts front-facing camera when the user holds their arm with with the palm oriented towards the user's face. Additionally, by adjusting their arm, users can use this camera as back-facing camera in order to record images and videos.

3.1 Screens

The device has two screens, although only one screen will be active at any given time. The device would have multitouch capability to support familiar gestures such as pinching and rotating. These gestures have become staples in modern devices such as tablets and smartphones. With the use of gyroscopes and accelerometers, the device is able to detect its orientation and active the screen that is most appropriate. The main screen appears on the top of the users forearm, on the same side most people wear their watches. The secondary screen is found on the bottom part of the forearm and will be visible whenever the users palm is oriented towards the user's face.

Most users are used to looking at the top of their forearm when interacting with wrist-worn devices such as watches or fitness trackers. As such, the main screen is where the user will spend most of their time interacting with the device. The main screen only has a **horizontal orientation**, since physically it would be uncomfortable to hold the device vertically and look at the screen at the same time. The main screen is where notifications will appear for the user. The user will be able to casually glance at the main screen to see notifications, making the same gesture used by people who are checking their watch. When the user wishes to interact with the device, the user can wake the device by pressing the Sleep/Wake button and unlocking the device. While interacting with the main screen, the user still looks at the device as if checking a watch.

The secondary screen is almost identical to the main screen. However, the constraint of the arm make it difficult for a user to hold their arm in such a way to see the secondary screen in a horizontal orientation. As a result, the secondary screen is limited to only a **vertical orientation**. Users will use the secondary screen when they wish to view something in the upright position or if they wish to use the device's camera as a front-facing camera.

The device has the menu interface that exists in tablets and smartphones. The main screen as well as the secondary screen have on-screen buttons which have a Back button, Home Button, and Search Button. When it came time to choose between capacitive and on-screen buttons, the design choice became on-screen for two reasons. One, since the the device two large screens, each one would need a set of capacitive buttons which would complicate the design. Two, capacitive buttons are always on, which make users prone to accidentally hitting the buttons when viewing videos or pressing buttons which are very close to the capacitive buttons. On-screen buttons, on the other hand, can be customized by users and positioned wherever the user chooses. This would allow right-handed and left-handed users to use the device with the same ease. In fact, the device's settings can be changed for right-handedness and left-handedness. In this way, the device would be as accessible to as many people as possible.

3.2 Hardware Buttons

The device has only three buttons. These are always found on the side of the device which faces the user's wrist. Most watches and wristwork devices already have buttons on the side of the device, so my design takes advantage of the cultural affordances associated with wrist-worn objects. The device has a Sleep/Wake button and two brightness buttons which are used to calibrate the brightness of the device. I chose this based on the physical constraint of using the device where only one hand is available. It is much easier for the opposite hand to access the side of the device closest to the wrist. It is much more comfortable for the user to access buttons this way, and doing so does not obstruct the screen, which is important when the user is trying to calibrate the screen's brightness.

The design of using hardware buttons came from need to having buttons for actions that must by easily accessible regardless of the state of the device. The choice for the Sleep/Wake button is obvious; all portable devices(to the best knowledge of the author) make use of this button. The brightness of the screen is equally important, especially in the case of a wearable device. In most, if not all, smartphones and smartwatches, the brightness settings can only be adjusted from the user's screen. That is, if the user needs to make the screen brighter because he has trouble seeing the screen, he must first navigate through the devices menus, go to the brightness settings, and then change the brightness. This is, frankly, ridiculous. By making the brightness of the screen accessible from two hardware buttons, the user can change the brightness of the screen at any moment without actually interacting with the screen itself.

The brightness buttons swap functions depending on the handedness of the individual. There is no confusion between the brightness buttons and the Sleep/Wake buttons due to their difference in shape. However, in order to stay consistent with the affordance of higher buttons meaning increase and lower buttons meaning decrease, the function of the brightness buttons must change depending on the arm on which the user is wearing the device. Obviously, the brightness button furthest from the interior of the users arm increases the brightness of the screen, while the button closest to the interior decreases the brightness.

4 Usage Scenarios

4.0.1 Notifications

Most smartwatches serve to send users notifications from their phones. This device can also alert the user to notifications on the main screen. When the device receives a notification, it vibrates twice in rapid succession in order to alert the user. However, the device will not wake up until it perceives that the user is looking at it. It does this by using its accelerator and gyroscope to detect if the user is holding their hand as if looking at their a watch. This is so the device is the least intrusive as possible. In this way, the device won't flash possibly sensitive information to other people. Once the user is looking at the device, the main screen becomes active and shows a queue of notifications that the user hasn't seen. If the user wishes, they can dismiss notifications by pressing the white x on the notification. If the user wishes to unlock the device, she can drag the key icon towards the lock on the other side of the screen and unlock the device. This lock screen is the same screen that a user sees right after Waking the device using the Sleep/Wake button. The only difference is that the the screen automatically activates when the device receives a notification.

The notification screen is split into two columns. The left column contains the notification that set off the notification. To right column has a queue of recent notifications.

The user can scroll on this queue to see rows containing icons for email, text messages, and other notifications. When the device wishes to notify the user of a notification, the device vibrates to let the user know a message was sent. The notification screen lights up

Here be the list

Now I will talk about why my device is awesome. It has a big screen, so now texting and surfing the internet will be a lot easier.

Talk about camera.

Talk about manual buttons.

5 Design Rationale

6 Advantages of Interface

7 Uses

8 Issues

9 User Response Predictions

Possible drawbacks: discomfort, super expensive, battery life, device getting really hot. Not being discrete.

A description of the type of system for which you have created the design, focusing on any particularly usability issues that youd like to address (see options in next section). 2. A top-level design or layout 3. At least two usage scenarios 4. Rationale for your design: relevant priorities, mental models, interaction design concepts, guidelines, principles, theories, etc. 5. Usability metric forecast analysis of your designif implemented then tested, what would