Designing Android Marshmallow Volume Controls: A User Experience Case Study

Noor Ali-Hasan

Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
nalihasan@google.com

Mindy Pereira

Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
mindyp@google.com

Rachel Garb

Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
rachelg@google.com

Abstract

When a user presses physical hardware volume keys on a smartphone, volume controls appears on the screen. Traditionally, these volume controls are represented by a single slider. Android's Lollipop release introduced new functionality into the volume controls: a way to temporarily silence interruptions from notifications and phone calls. After the launch of Android Lollipop, we discovered some usability issues with the more robust volume controls interface. We decided to address these issues in Android's next release, Marshmallow. In this case study, we describe the trade-offs we faced in designing Android Marshmallow volume controls and how our interdisciplinary user experience team designed a longitudinal study that helped us evaluate two designs we were considering. We also describe the impact of our research approach and how we arrived at a final design.

Author Keywords

Volume controls; Android Marshmallow; user experience; smartphones; mobile user experience; usability; longitudinal research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MobileHCI '16 Adjunct, September 06-09, 2016, Florence, Italy
© 2016 ACM. ISBN 978-1-4503-4413-5/16/09 \$15.00
DOI: <http://dx.doi.org/10.1145/2957265.2957273>

ACM Classification Keywords

Human-centered computing: Ubiquitous and mobile computing: Ubiquitous and mobile devices: Smartphones.

Introduction

When a user presses physical hardware volume keys on a smartphone, volume controls appears on the screen. Traditionally, these volume controls are represented by a single slider (see Figure 1). Android Lollipop introduced the addition of interruption controls to the volume controls interface in the form of three buttons: None, Priority, and All (see Figures 2 and 3). A smartphone set to None did not emit any sound at all. While the Priority option allowed the user to customize which interruptions could make sound; for example, the user could choose to have only calls from favorite contacts make sound. The All option permitted all interruptions (such as phone calls or push notifications) to make sounds.

Figure 1: The volume controls interface of Android Kitkat.

Figure 2: The volume controls interface of Android Lollipop.

After the launch of Android Lollipop, we discovered some usability issues with the volume controls

interface. We decided to address these issues in Android Marshmallow. In this case study, we describe the usability challenges and trade-offs we faced in designing Android Marshmallow volume controls and how our interdisciplinary user experience team designed a longitudinal study that helped us evaluate two designs we were considering. We also describe the impact of our research approach and how we arrived at a final design.

Figure 3: An expanded view of the None and Priority interruption controls of Android Lollipop.

Designing Volume Controls in Android Marshmallow

When we started rethinking the design of volume controls for Android Marshmallow, we examined all of the issues that users faced with the volume controls of Android Lollipop. We identified these issues through usability studies, feedback from Google employees trying the feature, and online feedback through the press and user forums. We describe below six key issues we faced and the solutions we identified for each issue.

Users didn't understand the interruption control buttons

We found that users did not understand what the interruption control buttons (none, priority, all) referred to or what they did. We decided to solve this issue by unifying these three buttons into one cohesive feature and giving that feature a name. We came to call this feature Do Not Disturb.

Users didn't know what Priority did

We also found that users were uncertain what would happen if they selected Priority. Users wanted to know what is considered a priority sound. To address this issue, we decided to educate users upfront about which sounds were considered priority by default and also offered users a direct link to a settings page where they could customize what is considered priority.

Users didn't like None

The None option silenced all interruptions, including alarms. This option did exactly as it was labeled but it didn't meet users' expectations. Users wanted a way to filter out all interruptions except for alarms. Most users we talked to told us that if they set an alarm, they wanted the alarm to go off under all circumstances.

We decided to give users exactly what they wanted. We replaced None with an option called Alarms Only that silenced all interruptions except for alarms.

Users wanted a way to silence everything

In Android Lollipop, we only offered the ability to silence interruptions, but not the sound effects and audio from games, videos, and websites. We considered those to not be interruptions since people were explicitly using them. However, users told us that they wished there was a way to guarantee that their device would be completely silent without turning it off. Users told us about wanting to use their phones in a quiet space where any sounds could be deemed disrespectful, such as at a movie theatre, wedding, or church. We added a new option called Total Silence that ensured that no sounds at all would be emitted from the phone.

Users wanted a way to access other types of volume

Android has more than one type of volume but we only show one volume slider at a time to keep the interface simple. The system picks the volume that seems most relevant to what the user is currently doing on their phone. Usually, that volume is the ringer volume, which controls the volume of phone calls and notifications. If a user is playing music or watching a video, they would be controlling media volume. But what if a user wanted to control a different type of volume like the volume of an alarm? In Android Lollipop, we did not offer an easy way to do it and had to use the Settings app. In Android Marshmallow, we offered users an expanded view (within volume controls) that quickly offered them a way to access other types of volume.

Does Do Not Disturb belong in volume controls?

Within our design and development team, we debated whether Do Not Disturb belonged in volume controls. On the one hand, we did see it as highly related to volume, and we felt that the Do Not Disturb feature was very valuable and something that users would want to use a lot once they found out about it. On the other hand, we recognized that not everyone would be interested in using Do Not Disturb, and it did add clutter to the volume controls for 100% of our users.

To resolve this point of contention, we designed two models: the integrated model and the separated model (see Figure 4). The integrated model makes interruption controls part of volume controls, while the separated model separates interruption controls from volume controls and places it in quick settings.

Figure 4: The integrated model (left) and the separated model (right).

Designing the Longitudinal Research Study

Traditionally at Google (as is typical elsewhere in the industry), a lot of our usability research tends to be lab-based. In a lab-based study, a user comes in for forty-five minutes to an hour to review or interact with a design product. There are many advantages to the traditional lab-based usability study approach. Lab studies are fast and allow us to iterate on our designs quickly. However, that approach usually only provides us with first impressions: how someone would initially learn to use a product. We have all probably experienced a product or user interface that was hard or confusing to use at first but then we grew to love it or see the merits of the design. With our new volume control designs, we were concerned that our lab studies were only providing us with users' first impressions. Moreover, with a feature like volume, we felt that it would be best evaluated if users could actually live with it for a while. There are certain volume scenarios and frustrations that we simply cannot simulate in a lab. We cannot simulate the frustration and panic of missing an alarm or the embarrassment of having your phone make a sound in the middle of an important meeting or event.

To account for these issues, we decided to conduct a longitudinal study of the two volume control models we were debating. We asked fifteen participants to live with the two volume control models we had designed. We provided participants with Nexus 5 phones running a special build of Android Lollipop. Participants swapped out their SIM cards from their current phones and lived with our phones for two weeks – one week trying out the integrated model and a second week trying out the separated model. To account for order effects (where research participants prefer something

based on when it was presented to them), half of the participants tried the integrated model first, while the other half tried the separated model. We also wanted to evaluate two names for the feature: Block Interruptions & Do Not Disturb. In the integrated model, the feature was called Block Interruptions and in the separated model it was called Do Not Disturb.

Developing the Prototypes

Our longitudinal study raised multiple new challenges for UX engineering. Previously, prototypes had been limited to use in a traditional usability lab setting and focused on a single task. However, volume controls are used in a variety of applications and scenarios. In order to determine the efficacy of each of the two models, we actually needed to create a solution that would allow our participants to completely replace their primary devices. The prototypes needed to be fully functional, allowing users to really feel as if they owned the device in hand; that meant everything had to work as expected, from setting the wallpaper, to watching YouTube videos, to accessing the participants own contacts and data.

Additionally, the volume controls in Android are part of the core system operating system. It was not technically possible to just create an application and install it.

To meet quality and experience expectations, we created a new branch of the Android OS based on the Android Lollipop release. This meant that participants did not need to learn an entire new platform, and removed focus from anything else that might be different on the phone. After that, the team expanded the volume control panel, quick settings panel, device settings, notifications, and audio manager services to

meet the new specifications for the prototype. We tested this build extensively internally by using it on our personal devices to ensure a high quality experience.

Given that these had become our participants' everyday devices, we had to make it possible for participants to switch between the integrated and separated model on the phones they were using. We added a hidden developer setting, out of sight from participants, that swapped between the two possible models at all the UI touchpoints, and in the audio and notifications surfaces. Our researchers provided the participants with instructions on how to switch between the two models in the middle of the study.

Research Results

Comparing the two models

When we do this type of research (comparing two designs), there is usually a need for the research to declare a "winner." In our experience, that rarely happens. If a team is debating between two designs, it usually means that there are merits to both. We found the same to be true in our study. Some participants preferred the integrated model, while others preferred the separated model. Participants who preferred the integrated model really liked the feature and found value in it, which prompted them to prefer the integrated model since it provided easiest and quickest access to the feature. They also found it cumbersome that in the separated model, they could not check their Do Not Disturb settings without turning it off and then back on (see Figure 5). Participants who preferred the separated model felt that quick settings was a more natural place for Do Not Disturb. They also felt that the

integrated model took up too much space in the volume panel.

Figure 5: To confirm their Do Not Disturb settings in the separated model, users have to turn off Do Not Disturb and then turn it back on to confirm their settings.

Comparing the two names

As part of this study, we also wanted to evaluate two names we had proposed for the feature: Block Interruptions vs. Do Not Disturb. Most of our study participants preferred Do Not Disturb. They felt that it more accurately described the feature. Some participants likened it to the Do Not Disturb door hangers at hotels.

A Third Model

We used the findings from our research to come up with a third model that combined the best of both models. We decided to place Do Not Disturb in quick settings, keeping it separate from volume controls like in the separated model. We did this to keep the volume controls as clean and minimal as possible. It was a more conservative choice, but when it comes to

something as fundamental as volume controls, we have learned that bold moves can be tricky. The tradeoff with this decision is that it makes Do Not Disturb a little less prominent. That means that it will be a little harder for users to discover and less accessible for those who want to frequently use it.

To improve discoverability, we added a shortcut to Do Not Disturb in the volume controls user interface. When a user holds down the physical hardware volume buttons, it eventually turns on Do Not Disturb, set to Alarms Only until the user turns it off. We have found this setting to be the most popular for Do Not Disturb and is very similar to how earlier Android phones would behave when they were put into silent mode. In that way, users could think of Do Not Disturb - Alarms Only as the new silent. By adding this shortcut, we essentially kept Do Not Disturb in the volume controls user interface as was the case with the integrated model.

Impact of the Design & Research Process

When we had conducted traditional lab studies in the past, there were times when various stakeholders in the team had doubts about some of the research findings. It is sometimes easy to dismiss usability findings when a user only used a feature for one hour. A user may have had trouble with a feature in the lab but how would they feel about it after using it for a few hours, a day, or even a week? Unlike in a traditional lab study, the longitudinal approach allowed us to understand the pros and cons of our two designs over two weeks (not just one hour). It also allowed our designs to live in “the wild” with users as they navigated real world volume scenarios (not just tasks in a usability study) like setting an alarm to wake up in

time for class or ensuring that a phone will not emit a sound in a movie theatre. Due to the thoroughness of our longitudinal study, various stakeholders in our team could not dispute the findings of the research. Moreover, the findings from the research gave everyone on our team a common understanding of the tradeoffs we were facing with the two models.

Conclusion

In this case study, we described the design and research process we took to arrive at the final design of volume controls in Android Marshmallow. This process marked the first time in our organization where we conducted a longitudinal study to understand product usability. The success of the longitudinal study required close collaboration between interaction designers, visual designers, user researchers, and UX engineers.

Acknowledgements

The authors would like to thank Allyson Tong for her contributions to the visual design of both volume control models, Francois Chu for his continuous feedback on our research plan, and Tom Karlo and Glen Murphy for their support of this research. The authors would also like to thank our research participants for giving up their smartphones for two weeks and for providing us with invaluable feedback about our two volume control models.