Case Study: Althea Takes on Ultimate Coder Challenge and Optimizes Shufflr* for the Ultrabook™

Althea Systems, creator of the Shufflr social video discovery app, recently updated the app for use with Microsoft's forthcoming Windows 8* operating system and Ultrabook™ devices. These updates were consistent with Althea's philosophy of ensuring cross-platform functionality for all users. Both updates presented opportunities to optimize Shufflr for the best possible user experience.

The Ultrabook update took place over just 6 weeks as part of the Intel® Ultimate Coder: Ultrabook™ Challenge, which Althea won. A large part of this development was focused on effectively incorporating the design device controls, such as sensors and touchscreen, and providing a user experience qualitatively different from tablets, smartphones, and traditional laptops. Althea's design team wanted to ensure that the user experience remained consistent between platforms yet integrated the best of each.

This case study describes the development opportunities and decisions that the Althea team made as they ported Shufflr to and redesigned it for the Windows Store app user interface (UI) and optimized it for Ultrabook. Key opportunities included:

- ❑ Redesigning Shufflr for a completely new operating system and learning which design components work best for devices that use it
- ❑ Ensuring that app controls were consistent from one operating system to another, so users can easily move between them
- ❑ Choosing the best programming language
- ❑ Making optimal use of specific device features
- ❑ Providing a high-quality user experience in the new app environment.

The Company

Althea creates products to enrich the online video experience on users' mobile phones, tablets, notebooks, PCs, TVs, and other devices. The company combines a vision to improve the way consumers experience online video with the latest technology and social components.

Althea's flagship product is Shufflr, a social video discovery app with versions for Apple iPhone* and iPad*, Google Android*, Facebook, and now Windows 8 and Ultrabook. Shufflr makes video suggestions based on each user's social media footprint and viewing history. The app enables users to see what their friends have been watching, and the buzz feed keeps them informed about what's trending on the Web. Shufflr also provides localized content and brings a personalized experience to online video.

The Product

Sagar Vinnakota, cofounder and vice president of engineering at Althea, describes Shufflr in more depth: "Users might be seeing videos on YouTube, Hulu, network television sites, or other online sources. To keep up with their favorite content, they have to seek out videos on each of those sites. But there are billions of videos out there, so how do users find the 10 or so they really want to watch each day? Shufflr does the work for them, actually delivering videos in a personalized way. Essentially, Shufflr lets videos find users, instead of the other way around, so users have an experience of serendipitously discovering videos."

Vinnakota and the Althea development team believe that this curation-style functionality is the best way to find, watch, and talk about online media (see Figure 1). Users can come to it from a variety of technology sources, including tablets, desktops, and smartphones. Social media also plays a role in how Shufflr works, because users can set it up to fetch videos from their private Twitter and Facebook accounts as well as public sites. If Facebook friends are recommending a video to a user, it will show up in his or her Shufflr interface. Vinnakota says, "All these access points become discovery points for users."

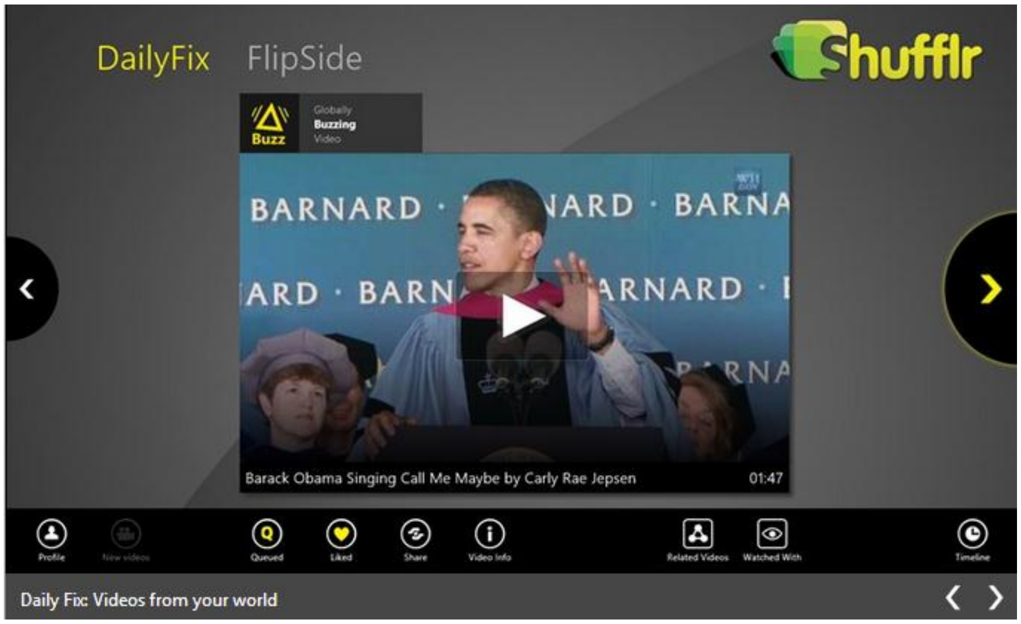


Figure 1. *The Shufflr UI*

Porting to the Windows Store App UI

In an effort to continue its cross-platform development, the Althea team wanted to create a version of Shufflr for Microsoft's forthcoming Windows Store. In doing so, they emphasized the importance of customizing the new version to leverage the hardware and operating system of the platform, reflecting a sound business strategy and good app development.

Vinnakota explains, "Shufflr is a screen-agnostic app, so we want to be everywhere our users are. We think Windows 8 will become a dominant force after it is released. Windows 8 Metro is completely fresh, a divergence from what Microsoft has previously done. It is completely different from Android or [Apple] iOS. So, it's not a matter of coding but more a situation of going back to the design board and back to developing a philosophy for what will work and what will not work."

Because of this divergence, the team was tasked with ensuring that the philosophy they developed would work—in practice—with Windows 8. Until the testing phase, several questions would remain unresolved:

- How does the newly designed app behave in the Windows Store app UI?
- Does the app perform functions in the same way with Windows 8 as with other operating systems?
- Do any components of the app need to be rewritten or revisited?
- How best can we optimize for Windows 8?

Development Process for Windows 8

Prior to starting development, the Althea team agreed on key success criteria. Vinnakota explains: "The key criteria for any design is to send the core message of the product to the user in an uncluttered manner, and then ensure that the experience is the most natural for a variety of use cases. In other words, it's important to not let design get in the way of a user experiencing the product. In this case, we wanted to ensure that users could spend time watching videos from their world of friends, tastes, and trends instead of getting caught up in searching for videos or navigating through an inordinate number of user interface levels."

Preserving the User Experience

The Althea design paradigm when moving Shufflr from one platform to another is to carry over the rich user experience while ensuring that it merges well with native platforms. For example, explains Vinnakota, "When Shufflr was brought to Windows 8, we had to ensure that controls that were previously overlaid on top of or near the video player were moved down to the app bar but still easily accessible for Ultrabook users. We intentionally did this in a way that shouldn't surprise Shufflr's Android or iOS users when they use the Windows 8 version."

Windows 8 Store Compatibility

The team also wanted to make sure the app design flowed naturally with the Windows Store app UI concept (ensuring priority of "content over chrome"). For example, to help ensure compatibility of the Windows Store app user experience, the team moved most of the context-sensitive and context-independent controls to the app bar.

When the team felt strongly that their design philosophy worked better than the standard Windows Store app UI guidelines, they broke the rules. For instance, Vinnakota justifies the prominent placement of left and right arrow controls at the edges to serve as navigation for previous and next videos: "We felt these were required, in spite of the multitouch screen swipe left–right controls for the same functionality. While such explicit navigation schemes that mix content and chrome are normally frowned upon in Metro user interface guidelines, it was clear for us that our design gives a better user experience in this specific case."

In developing for Windows 8, explains Vinnakota, the team had to convince Microsoft that what they were doing was compatible with Microsoft's design guidelines. He explains, "Microsoft has a very strong view on how Metro apps should look. So we worked closely with them, and we tried to convince them that while we appreciate the design interests of Microsoft and their philosophy, we have our own philosophy of what we would like our users to experience when it comes to online video discovery. Ultimately, Microsoft told us 'It's good that you broke a few rules here and there.'"

Another example is that most video content is in Adobe Flash* format, but Windows 8 does not currently support Flash. Vinnakota says, "We had to basically redesign the app. Shufflr uses a platform on the cloud that runs the algorithm that sends relevant content to users. So, we had to customize the behavior of Shufflr to show only non-Flash content. This means that a significant portion of the video content out there has to be converted before it is usable in the Windows 8 Store version of the app." Rather than doing content transcoding or conversion, Shufflr relies on open APIs provided by content sources such as YouTube, Vimeo, Dailymotion, and a number of premium content sources to provide videos that are compatible with each platform on which Shufflr is available.

Coding

Vinnakota notes, "We did not port the code base from iOS to Windows 8. The design and programming philosophies are entirely different in these platforms. At the most basic level, we did a native app for iOS, and we chose HTML+[JavaScript*]+CSS for the Windows 8 Metro app. The degree of flexibility offered by native code is not easily matched by an HTML+[JavaScript+]CSS setup. Yet, from a long-term perspective, it made sense for us to take this route on Windows 8."

The other two options for the Althea team were C++ and C#. In both cases, they thought they would be tied down to platform-specific implementation, whereas using a JavaScript+HTML approach to HTML5 was more portable to other platforms and hence flexible. Vinnakota observes, “Given that Internet Explorer* 10 (the underlying HTML5 engine) has full hardware acceleration support from the [operating system], we did not need to trade performance for portability.”

Althea’s design process involves several steps, including information architecture, concept drafting, paper prototyping, grid layout, UI language, and wireframing. Vinnakota notes, “In this redesign, the app design had to pass through all those stages and iterate a couple of times internally before the product reached the Windows Store.”

Development Process for Ultrabook

Following the Windows 8 redesign, the team had another opportunity to reinvent Shufflr: Intel invited Althea to participate in its Intel® Ultimate Coder: Ultrabook™ Challenge. In this contest, six developers competed for 6 weeks to create apps that take full advantage of the performance advances, graphic excellence, and touch and sensor technologies of the latest Ultrabook computers.

The process for creating an Ultrabook-friendly version of Shufflr was a learning opportunity for the Althea team. They realized that an Ultrabook has sensors, and many Ultrabook computers have a multitouch screen. Incorporating the touchscreen was obvious, but, Vinnakota says, “Originally we weren’t clear about why we would want to do anything with sensors on a laptop. Up to that point, we had not seen anything like them, so we weren’t sure what would make sense. Initially, we were shown models of Ultrabook where the screen is attached and behaves like a tablet. Or it folds, and the entire Ultrabook looks like a tablet. In both scenarios, there is a reasonably strong use case for some sensors, so we went ahead and started integrating the sensor code into Shufflr static content.” The team recognized that touch and sensors make sense on laptops but had to shift the way they thought about the app experience.

Sensor-based Features

Ultrabook offers a wide range of sensors and a 5-point multitouch screen to developers. From a design perspective, it was important for the team to pick and choose which ones to integrate into the Shufflr experience without making them look like a “force fit.” Vinnakota emphasizes that the team did not want to do something with sensors just because they are there. Rather, they thought through strong use cases, examined each sensor, and ran the sample test code to find out how each one works (see Figure 2).

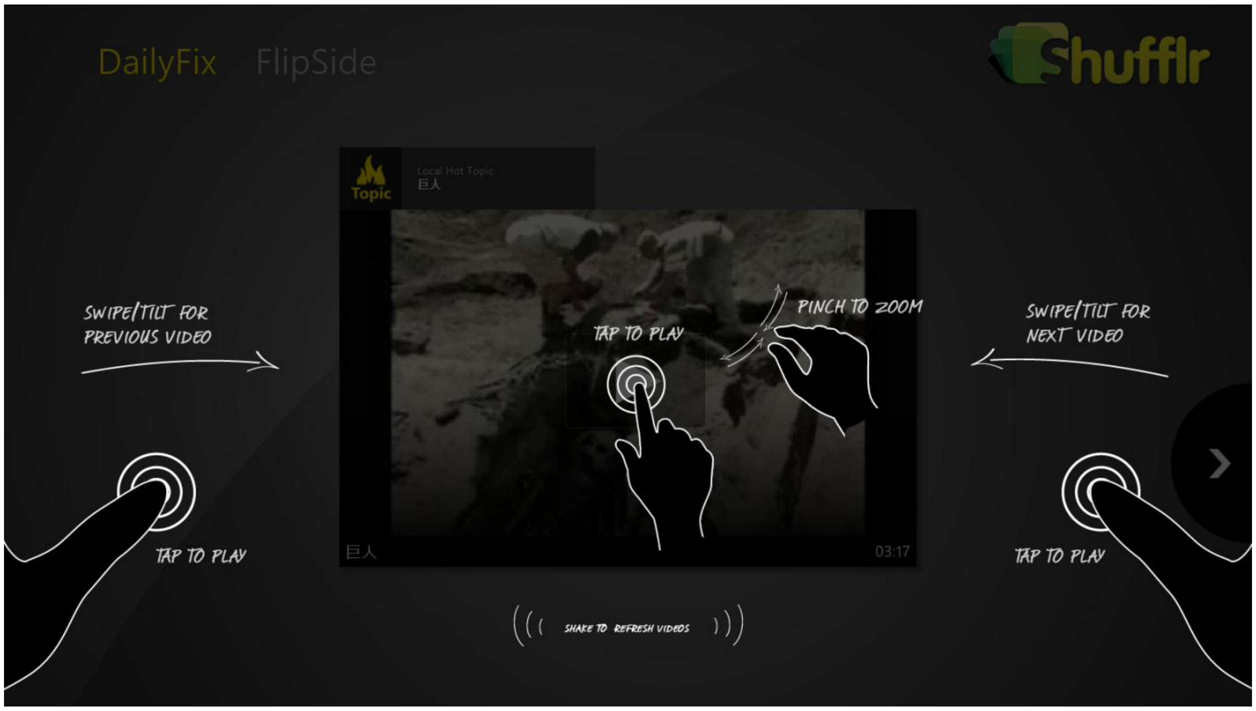


Figure 2. *Shufflr for Ultrabook™ was developed with strong use cases in mind.*

Navigation

In the current upgrade, Shufflr supports navigation for previous and next videos based on Ultrabook internal meters. The choice of sensors as well as the thresholds of the sensor values used for navigation were a key consideration in the design process. During week 3 of the Intel® Ultimate Coder: Ultrabook™ Challenge, the Althea team struggled with getting tilt navigation functionality to work properly. The goal was to be able to tilt the device to navigate to the next video (see Figure 3).

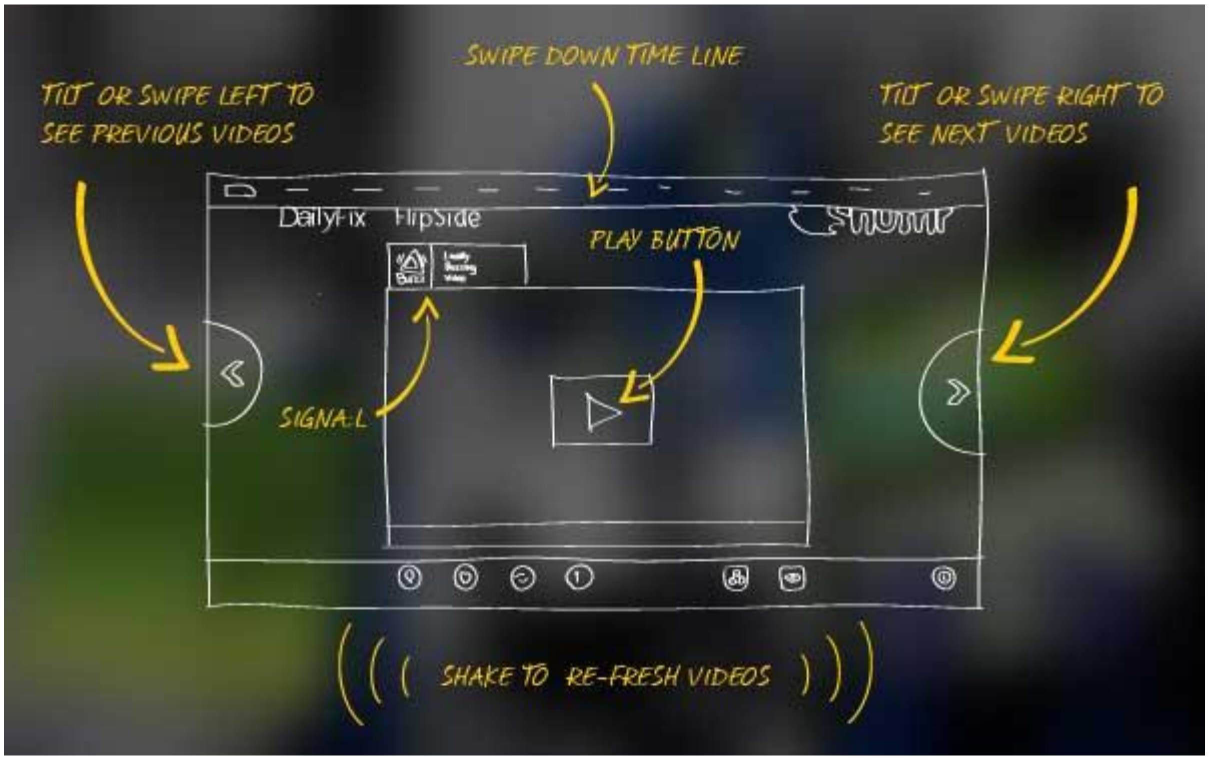


Figure 3. *Tilt and swipe design plan*

Initially, the team had the impression that the gyroscope was the sensor to use for incorporating tilt gestures. But at this stage, they also investigated the accelerometer. In an initial experiment, they used the Ultrabook to tilt, causing the next video to show up on the screen. After taking many measurements, they chose the g-force values on the *x*-axis for detecting left-tilt, at-rest, and right-tilt states for the Ultrabook. In the Althea blog, one of the team members writes that the goal was to "allow the user to get to the next and previous videos in the list without exaggerated movement." The post continues: "It should let the user be standing in a hallway, talking to a friend, holding the Ultrabook, shifting weight from one leg to the other, and not surprise him when he comes back to the app.[1]"

The team completed two designs and two implementations with the determined *x*-axis. According to the blog, the first design was based on "one video shift per tilt,[2]" which allows the user to move to a new video without using touch controls. The second design is based on "continuous shift in a tilt,[3]" which allows users to move through several videos in succession by maintaining the Ultrabook in a tilted position.

"The second implementation uses the 'roll' feedback from the Ultrabook inclinometer," states the blog.[4] The team observed that pitch and yaw were ignored during tilt navigation and surmised that "the inclinometer implementation should compete well with the pure accelerometer-based tilt navigation.[5]"

As of week 4 of the contest, both implementations were showing a glitch in “one video shift per tilt” mode, which the team concluded was the result of an issue in the tilt and rest detection state machine. This glitch would need to be fixed before the final release. At that point, the team wasn’t sure which implementation was better and left the decision for which one to use in the final design until a later time.

Ultimately, the team chose the inclinometer-based implementation. Vinnakota says, “We redesigned the tilt detection thresholds after collecting empirical data so that false positives and false negatives are minimized in the detection.”

Page Refresh

Shufflr for Ultrabook supports page refresh, using the accelerometers, through shaking. That is, if the user shakes the Ultrabook, the accelerometer detects this event and refreshes to show a new set of videos in Shufflr. The team faced a test when trying to integrate the internal accelerometers. Vinnakota explains, “Meters are native apps, so when the user is holding the device and walking around, the app must make the right call as to whether the readings or the values being given to the sensors should be responded to or ignored.”

Location-based Video Discovery

Vinnakota describes another function—the GPS sensor: “The new version also supports location-based video discovery. For example, a user who logs in from San Francisco gets a different set of local videos than someone who logs in from Bangalore or Tokyo.”

Always-on Video Updates

Another point of interest is an eye toward responsible green development—that is, taking power consumption into consideration without sacrificing the user experience. In the updated version of Shufflr, users are notified of new videos relevant to them when they are available. Using Intel® Smart Connect technology (which provides always-on, always-connected functionality), Shufflr enables users to stay informed of new video updates, even when their Ultrabook is in a low-power state. Vinnakota explains: “Shufflr does this by making a very lightweight call to its back-end server so that device power is not unduly consumed in that state.”

Ambient Light Sensor

The new version is still a work in progress, and Vinnakota comments, “We might include a proximity and ambient light sensor, as well. The use case for the ambient light sensor is based on the brightness of the room in which the user is accessing Shufflr. We would like the app to adjust the screen brightness accordingly. As a video app, users will spend a significant amount of time looking at the screen without taking their eyes off it, so we would like to integrate this feature."

This feature provides another example of a development opportunity. When a user's hand or some other obstruction, such as a piece of paper or a book, sets off the light sensor, it might adjust the brightness in a way the user did not intend. Vinnakota observes, "That would not be a very good experience for the user, so we had to find a way to prevent it."

White Eye

Another feature the team is looking at integrating is white eye, which needs native app support. Vinnakota says, "We still have to figure out a way to integrate white eye or a similar functionality into Shufflr."

Ultrabook can support both the light sensor and white eye, but the Windows Runtime (WinRT) cannot. Vinnakota explains, "Windows 8 native apps will support it, but there is a runtime called *WinRT,* which is what the Metro app runs on. So native apps can ultimately take advantage of ambient light sensors or white eye. At this point, it is something we are still investigating."

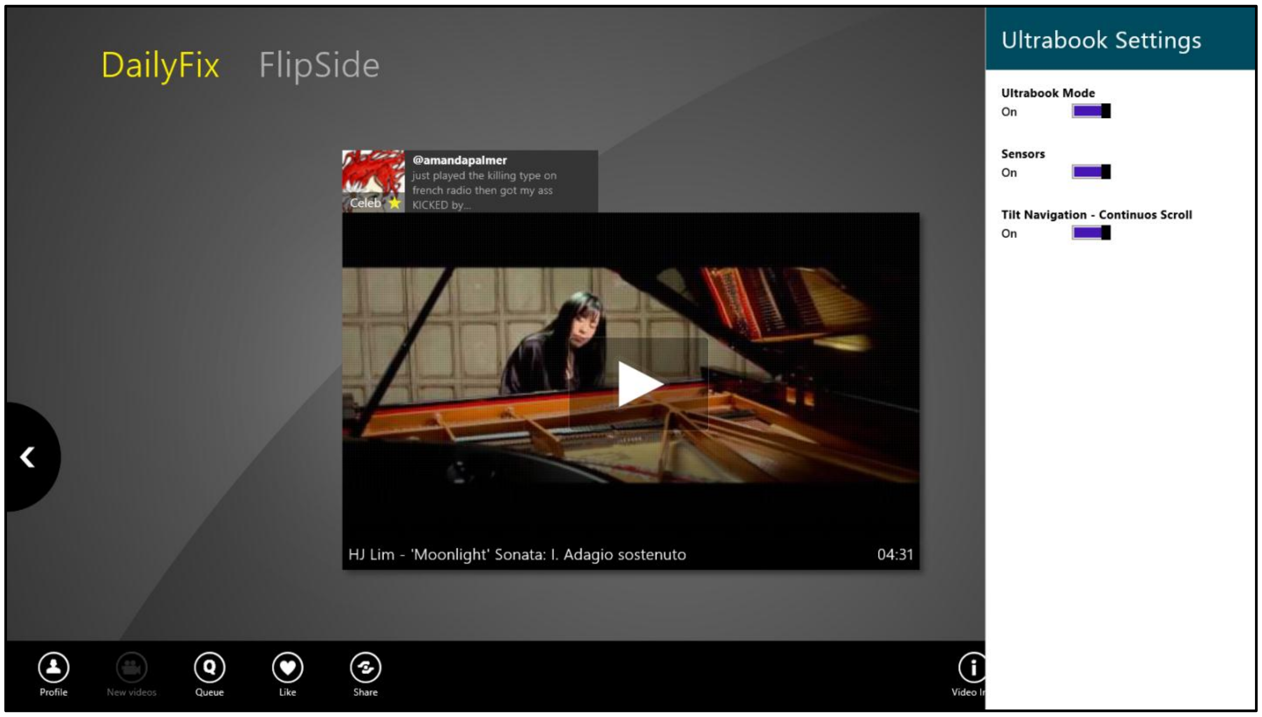
Touchscreen Optimization

Shufflr takes full advantage of the Ultrabook touchscreen, because most functions of the app can be accessed this way. For example, left swipe and right swipe are used to navigate to next and previous videos, respectively. The only need for a keyboard is when a user logs in or wants to type a comment when sharing a video. However, even logging in can be done through Facebook, where users have already entered login information that the app remembers.

Vinnakota notes, "We have designed our app primarily to be used with touch interface, because we have extensive experience developing touch interactivity. Users can operate the entire app with touch."

The team had to be sure and create visual feedback on various actions for touch users to ensure a smooth experience. For example, when users tap the **Next Video** button, they see a visual indication that they have already tapped it. Vinnakota explains, "Those kinds of things are expected by the experienced touch user."

The overall layout of the app as it appears on Ultrabook was also a consideration. Vinnakota says, "The design choices made for a 7-, 9-, or 10-inch tablet screen are not appropriate for a 13.3-inch Ultrabook form factor, because users cannot reach the entire screen with their thumbs while holding the Ultrabook. So, Shufflr has an 'Ultrabook mode' in which navigation and feature launches are closer to the edges (see Figure 4). Designing such a consistent, dynamic layout scheme needs considerable thought."


DailyFix
FlipSide
@amandapalmer
just played the killing type on french radio then got my ass KICKED by...
Celeb
HJ Lim - 'Moonlight' Sonata: I. Adagio sostenuto
04:31
Profile
New videos
Queue
Like
Share
Video I

Ultrabook Settings

Ultrabook Mode
On

Sensors
On

Tilt Navigation - Continuos Scroll
On

Figure 4. *Shufflr in "Ultrabook mode"*

Benefits of Ultrabook

Vinnakota and the development team saw several advantages to developing for Ultrabook. "For me," says Vinnakota, "one benefit was the pound factor. It's very light. It's also very fast. The time it takes to boot up and the time it takes to build a queue is very, very fast. The feature I like best is the touch. I think this is the next big thing."

The team saw working with Ultrabook as an opportunity to get an early look at technology that will gain strength in the coming months. Vinnakota explains, "We saw the Intel® Ultimate Coder: Ultrabook™ Challenge as an opportunity to get a very early look at the new devices, take them into our lab, and start experimenting."

And the Winner Is . . .

All their hard work paid off: Vinnakota and the Shufflr team won the Intel® Ultimate Coder: Ultrabook™ Challenge. The judges noted, "The Althea team was aggressive with high hopes of adding new features. But each week they hit roadblocks, and none of us were certain exactly where they landed. However, when it came to testing the app, it's apparent Althea has high standards. The judges were very impressed with the app experience, giving Shufflr the highest points for quality.[6]"

Some specific impressions included users understanding immediately that the app could be used differently from the way it is used on a standard laptop and the app's fit with Ultrabook and the Windows Store app UI. The judges commented about the contest as a whole: "Not only did we get great apps, but we saw innovation, technical solutions, new learning, and quality documentation that never existed 6 weeks ago.[7]"

For details on the winning entry, visit the Ultimate Coder Winner announcement.

Summary

Althea Systems, an app developer, recently participated in—and won— the Intel® Ultimate Coder: Ultrabook™ Challenge for its strong use of Ultrabook features and functions in a newly designed version of its flagship app, Shufflr. Prior to this contest, the Althea team had also updated Shufflr for Windows 8. These two updates presented multiple opportunities for the team to strengthen their cross-platform strategy and create a version of Shufflr for a whole new set of users.

Shufflr is a social video discovery app that finds and streams videos users might like based on their online footprint and stated preferences as well as their geography and social media streams. In taking the next steps toward becoming fully cross-functional among all platforms users might want to access, Althea decided to create a version of Shufflr customized to a Windows Store app UI. Because the Windows 8 environment is completely different from other platforms, the team felt it was necessary to redesign the app from scratch. Their goal was to deliver the core functionality of the app without allowing the design to get in the way of the user experience. Part of the opportunity to redevelop Shufflr was making key decisions about functionality that would fit this criterion. At times, these decisions conflicted with Microsoft guidelines for Windows Store app development, but ultimately Microsoft approved their decisions. The code decision was also based on the needs of the operating system; the team chose to work with HTML+JavaScript+CSS.

As with redesigning for Windows 8, decisions for the Ultrabook update were based on making the best use of the medium without forcing hardware features into the app design. Such decisions were crucial in optimizing sensor-based features such as navigation, page refresh, location-based discovery, and always-on functionality. As of the end of the contest, some features, such as an ambient light sensor and white eye, were still being considered.

Optimizing for the touchscreen was also critical in redesigning Shufflr for Ultrabook. Although previous versions of Shufflr made strong use of touch (with a few exceptions, the app can be used entirely by touch), the team had to ensure that the new version of the app's features and functions worked well with the larger screen.

In working with Ultrabook, the Althea team appreciated the device's lightweight form factor and fast speed as well as the integration of a touchscreen into a laptop.

1 vsagarv, “Shufflr and the Ultrabook Challenge: Week 3,” Althea Systems blog, August 27, 2012.

2 vsagarv, “Shufflr and the Ultrabook Challenge: Week 4,” Althea Systems blog, September 4, 2012.

3 vsagarv, “Shufflr and the Ultrabook Challenge: Week 4.”

4 vsagarv, “Shufflr and the Ultrabook Challenge: Week 4.”

5 vsagarv, “Shufflr and the Ultrabook Challenge: Week 4.”

6 Duffy, B., “Ultimate Coder Winner - Sagar Wins with Shufflr,” Intel Software blog, October 5, 2012.

7 Duffy, B., “Ultimate Coder Winner.”