공학석사 학위논문

Enhancing Mobile Interaction with Smartphone Cover

스마트폰 커버를 이용한 모바일 인터렉션

2017년 2월

서울대학교 대학원

컴퓨터공학부

최 고 은

Enhancing Mobile Interaction with Smartphone Cover

Advisor Jinwook Seo

Submitting a master's thesis of engineering

October 2016

Graduate School of Seoul National University

Department of Computer Science & Engineering

Seoul National University

Koeun Choi

Confirming the master's thesis written by Koeun Choi

December 2016

Chair _____Myung-Soo Kim_____ (Seal)

Vice Chair _____Jinwook Seo_____ (Seal)

Examiner _____Jehee Lee_____ (Seal)

Abstract

Enhancing Mobile Interaction with Smartphone Cover

Koeun Choi
School of Computer Science and Engineering
The Graduate School
Seoul National University

Most smartphones support multi-tasking with several means to switch between apps (e.g., a “recent apps” button or a “back” button). However, switching between apps is cumbersome when one has to do it frequently—for example, when notifications keep interrupting one’s current task. We introduce Peek-a-View, a fully transparent flipping screen cover that can reduce task switching overhead by providing an additional virtual screen space for subtasks. We assessed its feasibility in handling notifications. Upon receiving a notification, users can peek into the content of the notification without actually switching apps by slightly lifting the cover. If necessary, users can completely flip the cover to switch to the app that fired the notification. Two user studies showed that flipping and peeking interaction provided improved performance and proved to be useful for tasks that involve subtasks.

Also, we propose cover interaction input modalities based on cover angles. Two type events; state change which supports discrete flip cover state and angle movement

which changes value continuously when changing a flip cover angle; can be extended in future researches.

Keyword : Cover interaction; Mobile device; Task switching

Student Number : 2015-21268

Content

List of Figures

List of Tables

1. INTRODUCTION

Smartphones capable of performing multiple tasks at once are popular these days [1]. Multiple tasks include, but are not limited to, web browsing, video playback, social networking, and instant messaging (IM). Conventional multi-tasking in smartphones involved two steps to switch between apps: entering into app switching mode and selecting in the list of recent apps. Recent smartphone OSs (e.g., iOS 9 and Android Marshmallow) further introduced two techniques for multi-tasking: for permanent multi-tasking (Figure 1), one can split the screen space to display corresponding apps side-by-side; and for momentary multi-tasking (Figure 2), one can use a dedicated button to switch to a subtask and back out to the previous app upon completion of a subtask. However, both techniques have limitations coming from inefficiency of a smaller screen space for each app [16] and repeated task switching.

A response to a notification is one of the main cases that prompts immediate switching between multiple apps [17]. Instant messaging, for example, asks for feedback on a notification when users are engaged in other activities, such as watching a video. Upon receiving a notification, one has to determine whether it is worth an immediate response, and then switch to the app, and return to the original task. Such task switching causes high overhead to users [15], but a small pop-up on an upper status bar often does not provide enough information (e.g., first few words of an instant message besides a name or a profile photo of the sender) to determine whether the switching is necessary. Thus, users have to do explicit actions before making a decision (e.g., switching to the app that fired the notification or enlarging the status bar for more information). Such interactions have three limitations in terms of supporting efficient multi-tasking: (1) it requires multiple interactions such as entering an app switching mode or the home screen; (2) using a back button does not restore the original state in the previous app immediately, whereas notifications usually provide a “deep link” within an app (e.g., a particular chatroom); (3) the use of a back button is irreversible that switching again to the app that raised the notification requires additional interaction.

To streamline the task switching process, we propose Peek-a-View, which utilizes a transparent flip cover to introduce an additional virtual screen in multi-tasking scenarios.

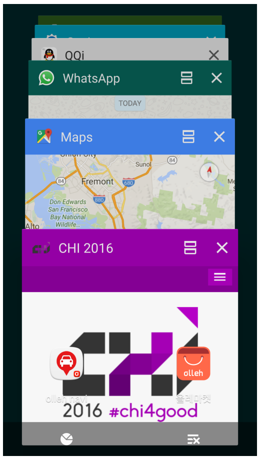

Figure 1. Task switching Method on Current Smartphones.

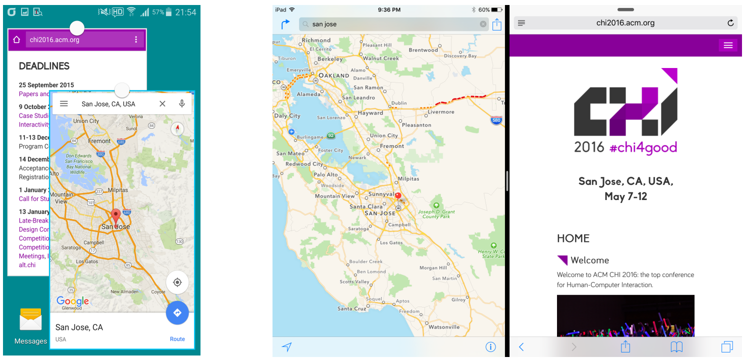

Figure 2. Multi-window on Galaxy Note 4 (left) and iPad (right).

2. RELATED WORK

In this section, we first review prior work that explores auxiliary interaction techniques for mobile input. Then we also summarize them as design space. We also introduce mobile device cover interaction including multi-layer display. Finally, we discuss possibilities of extending input on the device.

2.1. Auxiliary Input Interaction with mobile devices

Most smartphones adopt capacitive full touch screen. Touch commands by fingers are conducted as a major interaction input technique. Fingers block a portion of the smartphone's screen [10]. Moreover, the limitation of small sized screen of the smartphone makes users follow unnecessary complicated steps requiring users' multiple touch interaction when conducting tasks [16].

There are researches proposing additional complementary input techniques that help existing interaction efficiently using voice commands, platform-mapped gestures, built-in sensors, camera, external sensors, and wired/wireless communication modules with and without accessary. Table 1 shows possible auxiliary input method by hardware technology. Voice commands [26] and platform-mapped gestures are commonly supported by operator to replace or assist current input. Built-in sensors are studied by researchers actively. Wired/wireless communication modules such as USB, Wi-Fi or Bluetooth allow communication providing additional input/output from external devices or IoT things.

Voice commands [26] are commonly designed as additional input methods on wearable devices such as smartwatch and google glass because they have a limitation on touch interfaces by the physical size of the device. However, many users still feel uncomfortable using voice commands in public.

Gesture such as swipe, bezel swipe, or multi-touch gesture is considered as an assisted input method because of good accessibility and simplicity. Kim et al. [13] presented methods using bezel swipe to bring a challenges such as acquiring unreachable screen targets by one hand. However, bezel-swipe is commonly occupied by platform command and other gestures can be used by each application. This can make users confusing perform features.

Interaction methods using built-in magnetic sensor, such as MagiSign [11], MagPen [7], TMotion [27], MagiGuitar [12], and Magnetic Marionette demonstrate many novel interaction scenarios allowing detection of the orientation of a stylus, off-the screen interaction, pressure-sensitive input or playful forms of interaction with mobile phone. MagiSign [11] is a touchless, gesture-based authentication method based on the interaction between a magnet and a device. Hwang et al. [7] and Yoon at el. [27] enriched pen interaction as exploring the design space of a magnetically augmented stylus.

Butler at el. [2] investigated sensing user touch around small screens. They used infra-red (IR) proximity sensors capable of detecting the presence and position of fingers in the adjacent regions so that user can carry out gestures using the space around the device.

Possible Trigger Mechanisms for auxiliary Input	H/W Technology	Usage
Touch	Touch panel (Swiping)	Use in application
	Touch panel (Bezel Swipe)	Triggering platform menu
Voice	Microphone	Command [26]
Built-in Sensor	Proximity sensor	
	Gyro sensor	Tilting Gesture [3]
	Magnetic sensor	3D mobile input, stylus [7], [11], [12], [27]
Camera	Camera module	
External sensor	IR proximity sensor	In-air gesture [2], [10]
Wire/Wireless communication	USB/Wi-Fi/Bluetooth	IoT, external device communication

Table 1. Possible Auxiliary Input method by hardware Technology.

2.2. Cover Interaction & Multi-layer Interaction

Prior work on a smart device cover ranges from adopting it as an auxiliary input device with additional buttons on the cover [22] to using it as a secondary output for either display of relatively simple information [5] or full duplication of primary display contents with an e-ink panel [6]. Also InkCase [8], PopSLATE [18] and Flexcase [20] added secondary e-ink display on the back of mobile device case. Although these products show compelling scenarios, it has still limitation on functionality and performance on e-ink display. Rendl et al. [19], on the other hand, introduced a thin transparent sensing surface to be attached to a touch screen. It enriched interactions on the touch device by using deformation of the film as an additional interaction dimension.

Previous researches also explored multi-layer interaction using additional mobile devices on the tabletop [23, 24]. For mobile devices, Cauchard et al. [4] used directional information acquired from an additional sensor to switch between virtual workspaces. As an alternative method, one could utilize the internal proximity/gyro sensors or swiping gesture to switch the workspaces, but the techniques could suffer from unstable grip or interfere with existing gesture mapping. Peek-a-View uses a readily available fully transparent flip cover that does not require any additional sensors. Due to its transparency, users can see and interact with the entire touch screen with the cover shut as if there is no additional interaction medium.

3. DESIGN AND IMPLEMENTATION

As we reviewed, our work also contributes to enrich auxiliary input interaction research on mobile device. Currently, flip cover is one of mobile accessories. However, it has the possibility to being used as one of major input methods. Foldable mobile devices are about to launch on market. Mobile interaction with angle input can enrich additional input compensation tool with touch interface which is a main input method. In this section, we explore the input modalities the flip cover form factor affords and prototype implementation detail for one type of input modalities.

3.1. Input Modalities

Smartphone can detect angle using movement of a secondary display or a flip cover. We divide flip cover angle movement input to two different event types so that they can apply to various contexts (Table 2).

State Change. Users can perform different activities when changing their grips. When changing the cover being closed, peeked, opened, or flipped, applications are able to map different behavior by the states. We introduce Peek-a-View which is one use case to confirm good usability of cover interaction.

Angle Movement. While users perform a task, an application can respond continuous input to control a part of the task. It gives the possibility to assist some controls' accuracy, playful form of interaction.

Event	Input Type	Input Value	Usage
State Change	Discrete State	Closed, Peeked, Opened, Flipped	Mode Change
Angle Movement	Continuous Number	0^{o}~360^{o} or Converted number	Input Controller

Table 2. Input modalities for Cover Change.

3.2. Design and Implementation of Peek-a-View

3.2.1. Design

Peek-a-View adopts "virtual desktop" to address limited screen space for multi-tasking on mobile devices. Virtual desktop is a system used to expand virtual workspace beyond the physical limits of a screen's display area. It allows the user to switch between different screen spaces on a PC [21]. Our system is composed of two virtual spaces that the user can interact with. Each virtual space can be switched by flipping the cover. It enables users to perform tasks on explicitly separate workspaces. In addition to the *closed* (Figure 3a) and *opened* (Figure 3c) state of the cover, the combination of proximity and light sensors enabled a new peeked (Figure 3b) state. We assigned two virtual workspaces provided by *opened* and *closed* state to two different apps, and introduced "peek" interaction to show a brief information of the app on secondary display.

3.2.2. Implementation

The system was implemented with built-in sensors (i.e., light sensor and proximity sensor) in the mobile phone (Figure 4). We defined each state with the combination of sensor values (i.e., ***closed***: *proximity* = 0, *light* < threshold, ***peeked***: *proximity* = 0, *light* > threshold, ***opened***: *proximity* > 0).

Prototype implementation supports running two applications on two virtual spaces by three modules; flip cover state broadcaster, notification listener and Peek-a-View controller. Flip cover state broadcaster broadcasts three states of a flip cover; *closed*, *peeked* and *opened* when moving a flip cover. Notification listener catches various notifications registered by a user. Peek-a-View controller handles showing and hiding applications by each states. It also supports a quick context of notification in *peeking* state which enables users to decide task switching easily (Figure 5). Peek-a-View can apply to any application and current notification messengers on the market (Figure 6).

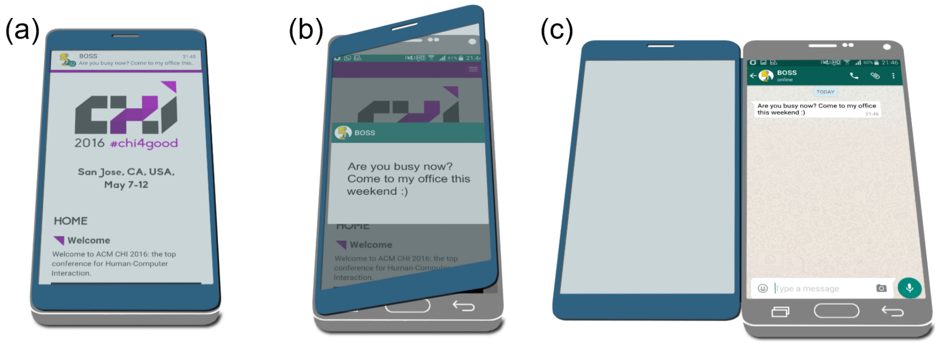

Figure 3. Three cover states: (a) Closed – showing App-A, (b) Peeked – showing quick context of App-B, (c) Opened – showing App-B.

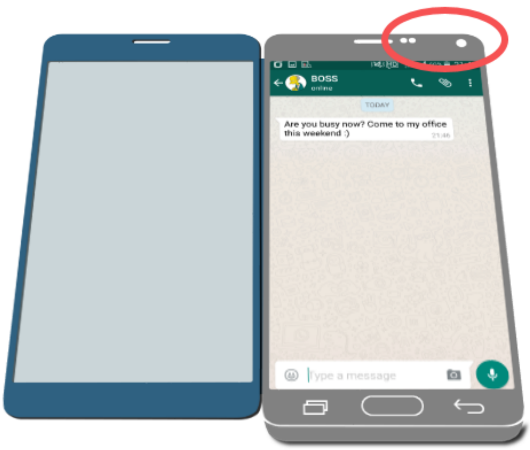

Figure 4. Proximity sensor and light sensor on Smartphone.

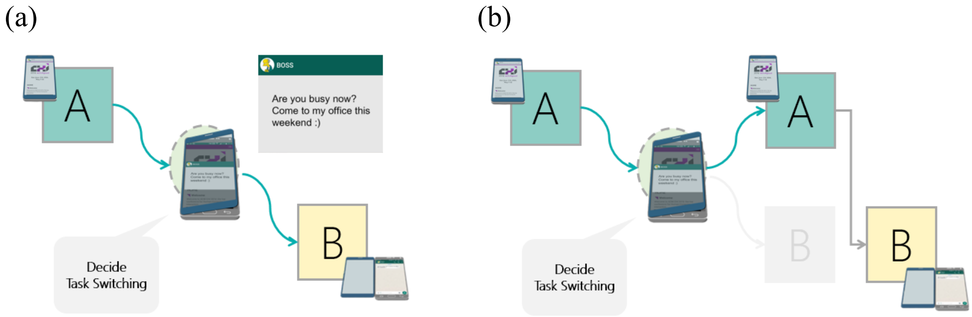

Figure 5. Task switching scenarios. Peek-a-View supports *peeking* state which enables users to decide task switching easily.

(a) Immediate task switching from task A to task B through peeking behavior.

(b) Scenario shows continuing task A without switching to task B immediately.

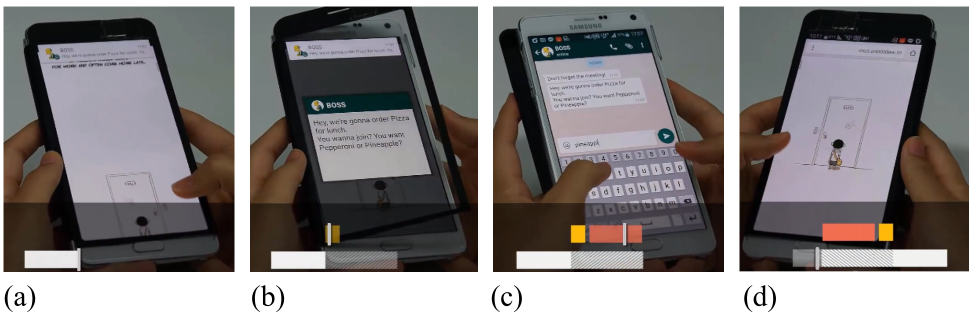

Figure 6. Peek-a-View. Pictures show scenario between web browsing task and messaging task using WhatsApp messenger.

(a) Notification comes when browsing. (b) User checks the notification by peeking gesture. (c) User chats using WhatsApp messenger by flipping a cover. (d) User performs original task by closing a cover.

4. USER STUDIES

4.1. Feasibility Study of Cover-Closed interaction

A study was designed to (1) examine the affordance of a cover interface and (2) check which of the two workspaces (a covered screen, Figure 3a, or a bare screen, Figure 3c) is more suitable for the main virtual display, while keeping the other for subtasks.

4.1.1. Design and Procedure

We prepared short passages of writing over various topics (each with 152±15 words). We extracted 18 passages from the Internet. After that, we measured their difficulties through an online-survey participated by 15 volunteers aged 20 to 38 years (M = 30, SD = 4.59) based on criteria made by Jo et al. [9]. We used six easy and one difficult passages from [9] for the difficulty baseline. After sorting the passages, we picked the 12 easiest passages out of the 18, which we combined with the six easy baseline passages to make a total of 18 passages for the experiment.

We investigated task switching scenarios using notifications (Table 3).

Notification Type	Message description
NT-N *(No interaction)*	Messages that participants do not need to answer as in Figures 7a and 7b. (e.g., "You have a new message. Please ignore this message.")
NT-O *(One interaction)*	Messages that participants MUST reply to immediately as in Figures 7c and 7d. (e.g., "You have a new message. Please put an answer of 3+4=?")
NT-M *(Multiple interactions)*	Messages that participants MUST reply to but they need to refer to the content from their current task as in Figures 7e and 7f. (e.g., "You have a new message. What is the character count in the first word of the 3rd line on a given passage?"

Table 3. Questions given for notification message task.

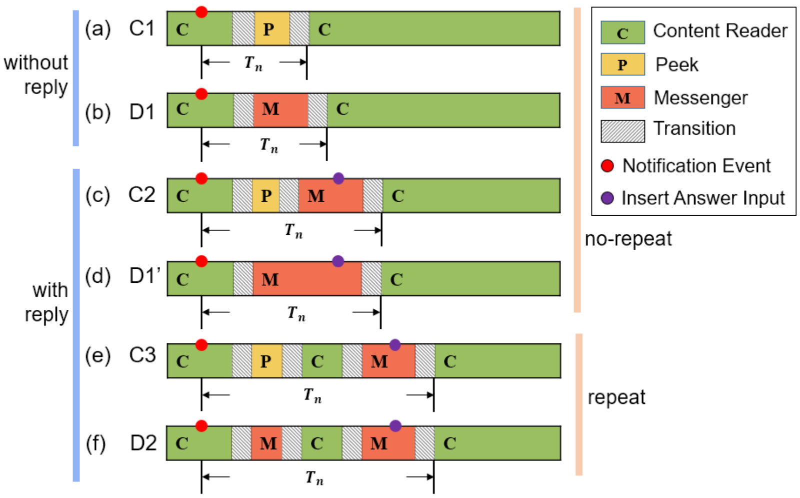

Figure 7. Response Task Switching Patterns. C1, C2, C3 means the pattern only using Peek-a-View. D1, D2 can be done with Peek-a-View (CO & CC) and Baseline (BS).

All messages were over two lines long, and the first line deliberately showed the same text for all the types to make sure that participants were not able to figure out the type from the notifications without reading the entire message.

We built two apps: Content Reader (Figure 8) and Messenger (Figure 9). Content Reader was composed of a text viewer and a "start" and "end" button. A pseudo-messaging app, which we designed after analyzing the top 5 most popular global mobile messenger apps (i.e., WhatsApp, QQMobile, WeChat, Skype, and Line) [25], delivered the notification messages. All five messengers required pressing a "back" key twice to exit the apps (once to leave a chatroom, and once again to exit from the applications). We excluded the Facebook Messenger and Viber because their popups block the viewport of the display unlike the others.

The participants performed the task with three interfaces: with a cover starting in flip-cover-closed state (CC); flip-cover-opened state (CO); and without a cover as a baseline (BS). BS includes the following existing task switching methods: using "back"; "home"; and "recent apps" buttons.

Each trial lasted for one minute. Only one notification, which appeared after 10 seconds into a trial, was received for each trial. All the three interfaces were performed six times (trials) (3 types × 2 repeated), and 18 passages were assigned to each of the 18 trials using a balanced Latin square. Once the participants pressed a "start" button, they were told to read a passage. We asked them to check the message upon receiving them, take appropriate action, and return to the original task to continue reading the passage. When they pressed the "end" button, a question popped up asking about the content of the passage, providing three choices (a correct answer, a wrong answer, and "I don't know"). Upon selection of a choice, the next trial started.

A three-minute training session was conducted for each method. After all the trials, we collected feedback using a seven-point Likert scale questionnaire (1=strongly disagree, 7=strongly agree), followed by an interview. A questionnaire asked for learnability, ease of use, comfort level, and satisfaction.

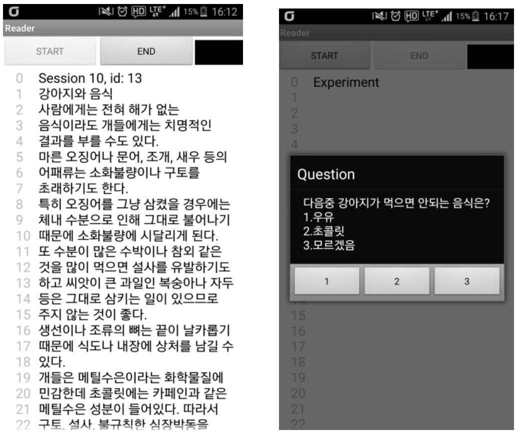

Figure 8. Content Reader Application

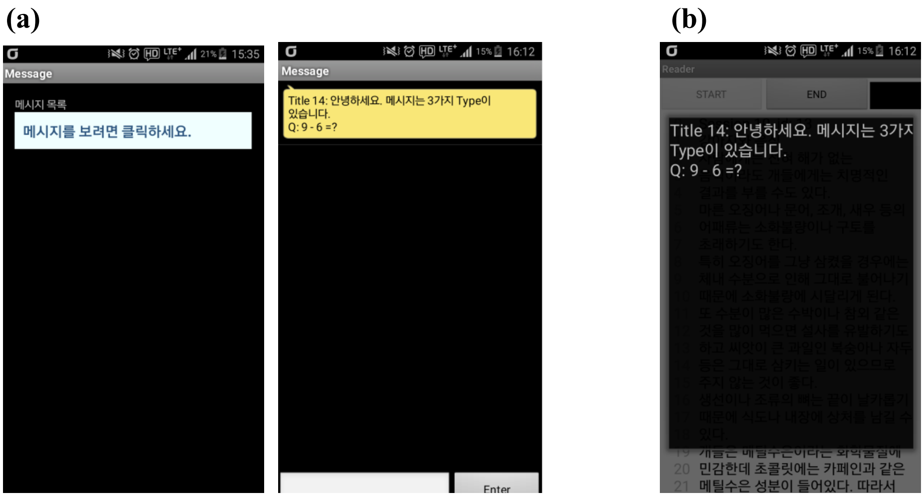

Figure 9. (a) Messenger list display and chatroom (Left),

(b) Message Popup for Peeked State (Right)

Question	BS	CC	CO
Q1. This interface is easy to learn.	5.8	5.6	5.4
Q2. This interface is easy to switch tasks.	3.9	6.4	5.8
Q3. This interface is comfortable to use.	4.5	5.9	5.1
Q4. I would like to user this interface again.	4.5	6.0	4.6

Table 4. Average Likert scale ratings for the three interfaces using the scale of 1=Strongly disagree and 7=Strongly agree.

4.1.2. Experiment Setup

Each participant worked on Samsung Galaxy Note 4 and our prototype in controlled laboratory. For controlled study, phone was set to airplane mode and notification access was allowed for Messenger application.

Automatic logger software collected response task completion time and error rates including response task answer and contents comprehension task answer per user and notification type.

4.1.3. Apparatus

Our prototype was built on a Samsung Galaxy Note 4 (153.5 × 78.6 × 8.5 mm) running Android 5.0. A fully transparent tempered glass flip cover preserved the clarity of the screen.

4.1.4. Participants

Eight volunteers (2 females and 6 males) from a university participated in the user study. They were all graduate students. 5 of them are majoring in Computer Science and Engineering, and others are from different majors. They were aged 20 to 38 years

(M = 29, SD = 4.70) and have used Samsung Galaxy series smartphones for over six months.

4.1.5. Results

We analyzed the results using two-way RM-ANOVA and found significant main effects on response task completion time by interface ($F_{2,14}$ = 5.8, p = .015) and notification type ($F_{2,14}$ = 19.728, $p < .001$) (Figure 10a). There were also significant interactions on interface and notification type ($F_{4,28}$ = 3.224, p = .027). Post-hoc tests using Tukey's pairwise comparisons showed a significantly shorter (p = .008) response time (sec) using CC (M = 8.215, SD = 0.696) than BS (M = 14.214, SD = 1.365). As predicted, there were significant differences among NT-N, NT-O, and NT-M by Tukey's pairwise comparisons (Figure 10a).

We also analyzed the seven-point Likert scare questionnaire results (Table 4, Figure 10b) using two-way RM-ANOVA. Easiness to use and satisfaction were significantly different between CC and BS ($F_{2,14}$ = 22.980, $p < .001$ and $F_{2,14}$ = 4.057, p = .041, respectively), whereas there were no significant effects on learnability and comfort level. Post-hoc tests using Tukey showed that ease of use and comfort level for CC were significantly higher than BS (p = .020).Our prototype was built on a Samsung Galaxy Note 4 (153.5 × 78.6 × 8.5 mm) running Android 5.0. A fully transparent tempered glass flip cover preserved the clarity of the screen.

The study showed the better speed of cover interface than others, especially for CC. Moreover, there were meaningful results for CC on user preferences.

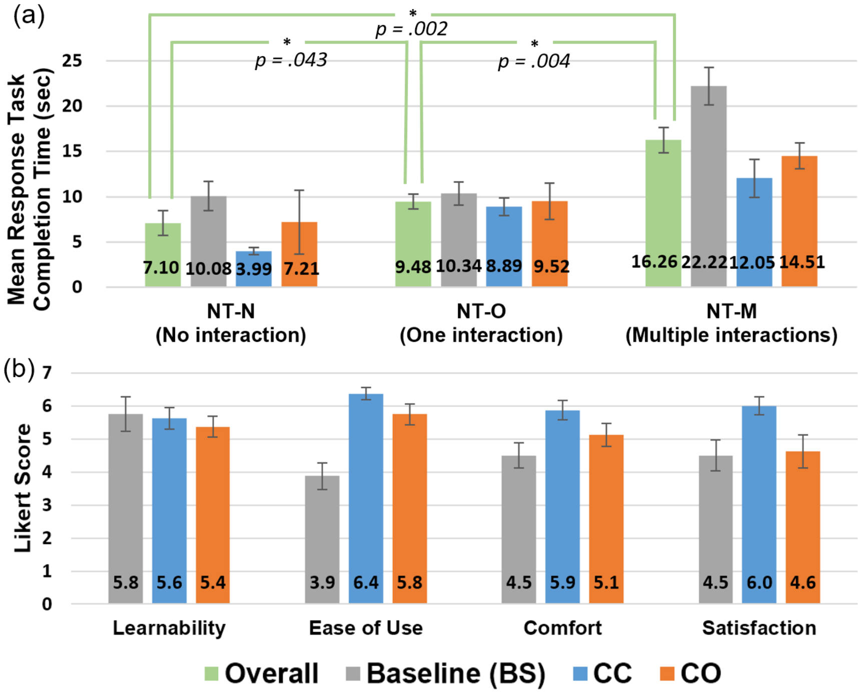

Figure 10. (a) Mean response task completion time and (b) mean questionnaire results. Error bars denote standard error.

4.2. In-depth Comparative User Study

Based on the findings from the previous study, we conducted a follow-up user study to further evaluate the efficacy of Peek-a-View using CC in supporting task switching.

4.2.1. Method

We prepared 18 passages and a device identical to the one in the previous study. We used a within-subjects design with two independent variables: interface (CC and BS) and notification type (NT-N, NT-O, and NT-M). Each participant performed three trials per condition, which resulted in $2 \times 3 \times 3 = 18$ trials per subject. For each trial, each participant was asked to do same task as the previous study. We measured the task completion time in each phase of the task. After the experiment, the participants filled out a questionnaire based on a modified SUS (System Usability Scale) [14] for subjective evaluation.

4.2.2. Experiment Setup

Each participant worked on Samsung Galaxy Note 4 and our prototype in controlled laboratory. For controlled study, phone was set to airplane mode and notification access was allowed for Messenger application.

Automatic logger software collected all time information including the task completion time, response time, task switching time, transition time, and error rates including response task answer and contents comprehension task answer per user and notification type.

4.2.3. Apparatus

Our prototype was built on a Samsung Galaxy Note 4 (153.5 × 78.6 × 8.5 mm) running Android 5.0. A fully transparent tempered glass flip cover preserved the clarity of the screen.

4.2.4. Participants

We recruited 18 participants (5 females and 13 males) from a university and a company. The participants were aged 20 to 34 years (*M* = 26, *SD* = 4.67), and all of them had experience with the Galaxy series smartphones for more than six months. They were compensated about US$10.

4.2.5. Results

We first evaluated efficiency in terms of task completion time and then analyzed the usability of the proposed interface using questionnaires. Since each task included primary and secondary tasks and transitions between the two, we measured every individual time span including the transition time. Then, we defined response task completion time, and further scrutinized the subsequence of the task. We used two-way RM-ANOVA in temporal analyses.

Overall task completion time (sec)

Overall task completion time was defined as the time span from the beginning till the end of each task. It included the time to finish both reading and response task. CC was faster (*M* = 45.6 and *M* = 47.9, respectively), but the difference was not significant.

Response task completion time (sec)

Response task completion time was defined as the time span from the arrival of the notification to the completion of transition after the response. We analyzed response task completion time (T_n in Figure 10) and found a significant main effect of interface ($F_{1,17}$ = 112.37, $p < .001$), and notification type ($F_{1.24,21.12}$= 15.98, $p < .001$). There was a significant interface × notification type interaction effect ($F_{1.25,21.25}$= 15.98, $p < .001$). For all the three notification types, CC was significantly faster than BS (Figure 11), and post-hoc tests using Bonferroni's pairwise comparisons showed significant differences between all the three notification type pairs ($p < .001$). However, the difference between CC and BS was relatively smaller in NT-O resulting in the significant interaction effect.

Notification recognition time (sec)

Since each response could be divided into recognition, preparation, and text entry, we further scrutinized the elapsed time for each action. We excluded NT-O notification for instant response as the latter two actions could not be distinguished when one responded to the notification immediately. There was no significant difference for preparation and text entry, but recognition time using CC was significantly faster ($F_{1, 17}$ = 119.06, p < .001) for both NT-N (M = 1.16 and M = 2.36, respectively) and NT-M (M = 1.97 and M = 2.86, respectively).

Transition time (sec)

Besides the elapsed time on both apps, we also measured transition time between them. We found a significant difference ($F_{1,17}$ = 106.135, p < .001) between CC and BS (M = 0.064 and M = 0.763, respectively). Notification types also showed a significant difference ($F_{1.44,24.43}$ = 19.642, p < .001) and post-hoc tests using Bonferroni's pairwise comparisons showed significant differences between NT-N and NT-M (M = 0.340 and M = 0.583, respectively, p < .001) and between NT-O and NT-M (M = 0.317 and M = 0.583, respectively, p = .001).

Peeking

[28] showed the average reading speed of text on a screen was 180 words per minute. Base on this, we speculated that it would take at least 333ms to read one word. The peeked state was used for 88.8% (*n = 142, M = 1.386 sec, SD = 0.611*) of all trials when we counted the occasion participants stayed in *peeked* state for longer than 333ms.

Usability

We analyzed the SUS rating for two interfaces using a paired-samples *t*-test (Table 5). There was a significant difference (t_{17} = -4.76, $p < .001$) between CC (M = 80.833, SD = 11.568) and BS (M = 67.083, SD = 11.056).

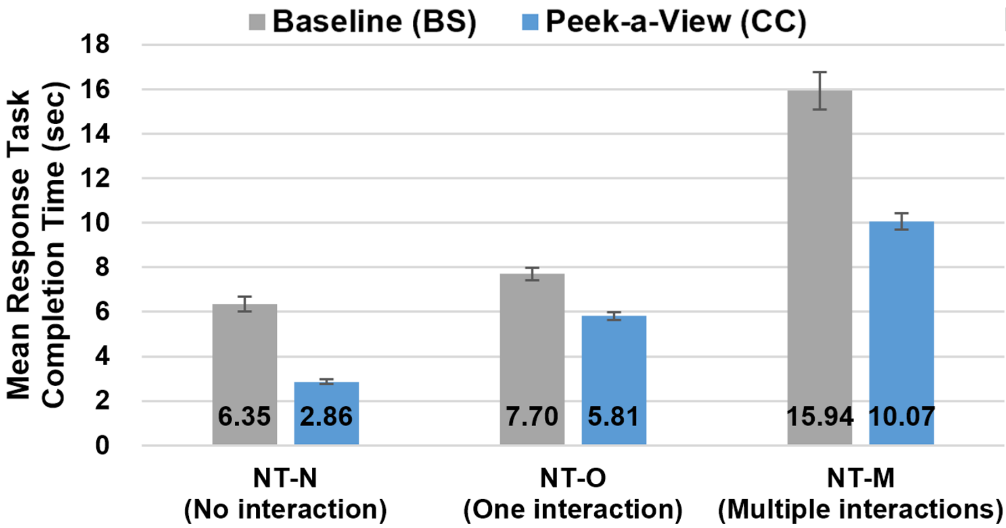

Figure 11. Mean response task completion. Error bars denote standard error.

Question	**CC**	**BS**
Q1. I think that I would like to use this product frequently.	**4.1**	3.3
Q2. I found the product unnecessarily complex.	**2.0**	2.7
Q3. I thought the product was easy to use.	**4.5**	3.4
Q4. I think that I would need the support of a technical person to be able to use this product.	2.0	**1.8**
Q5. I found the various functions in the product were well integrated.	**4.1**	3.3
Q6. I thought there was too much inconsistency in this product.	**1.8**	2.6
Q7. I imagine that most people would learn to use this product very quickly.	**4.4**	3.7
Q8. I found the product very awkward to use.	**2.0**	2.2
Q9. I felt very confident using the product.	**4.2**	3.7
Q10. I needed to learn a lot of things before I could get going with this product.	**1.8**	2.1
A mean SUS Score *	**80.8 (*SD*=11.6)**	67.1 (*SD*=11.1)

Table 5. Average Likert scale ratings (a Modified SUS questionnaire [14]) for the two interfaces using the scale of 1=Strongly disagree and 5=Strongly agree. Bold color presents better usability score.
A mean SUS Score with significant difference was marked with an asterisk (*).

5. DISCUSSION AND LIMITATION

Analysis on time spent for each action revealed that the superior efficiency of Peek-a-View came from faster recognition of notification content and faster transition time between apps. In the conventional condition, the participants showed mixed usage of the “back” and “recent apps” buttons for transition and barely used the “home” button to switch apps. While conventional methods involve two interactions (i.e., one for entering a switch mode and the other for selecting an app), Peek-a-View needs only one physical interaction. It provides peek capability to check the content quickly with a single hand. However, more than half of the participants used both hands to flip the cover completely. It depicts the difficulty of single-handed flipping, but the interaction usually involves performing intensive tasks such as keyboard typing and non-casual browsing, in which case the use of both hands is advantageous. It showed strength in frequent app switching with its reversibility compared with the use of “back” button. In particular, the participants had to use the “recent apps” button for NT-M because of the need for multiple switching. Along with efficiency, analysis on the SUS rating also showed that the flip cover interface was more preferred by the participants. It could be attributed to familiarity with the flip cover as majority of the participants (77%, $n = 14$) have used or are using the cover. Three participants said that it would be great if this feature is applied to KakaoTalk and Facebook. On the other hand, one of participants said that he did not like the accessary so that it needs more attractiveness for cover for him.

Except for one participant who did not actively use the *peeked* state, the utilization was at 92% ($n=141$). P7 and P14 said it was useful for NT-N and NT-M, but not so much for NT-O because they had to open the cover eventually to respond. NT-N and NT-M would benefit from the *peeked* state, because a natural follow-up action for both types would be to come back to the original task after checking the message in the *peeked* state, whereas NT-O would require entering the subtask. However, since the *peeked* state is the transitional state into the *opened* state considering the movement of the cover, the penalty would be at its minimal.

In our prototype, we relied on proximity and light sensor to detect the state of the flip cover. It showed feasibility even with the existing features of smartphones. While the number of states is limited to three in the current implementation, the three states were proved to be sufficient for multi-tasking between two apps. Also, P8 and P11 said that they wanted to see more notifications than just the most recent one. A combination of additional magnetic and other sensors can be used to detect the angle of the cover to achieve finer granularity and introduce a higher number of states for task switching between more than two applications. Furthermore, it can enrich the type of interactions more than a "peek".

There are design alternatives without using a flip cover. When we explored the design space, we considered many alternatives; the user of a proximity sensor, a gyro sensor, a swiping gesture, and an extended notification bar. However, they also have limitations compared to ours.

Proximity sensor: As the sensor is typically located on the top of the screen, one's interaction with the sensor not only occludes the screen but also results in an unstable grip when one tries to reach it with a single hand. On the contrary, one can easily "peek" even with a single hand without changing a grip. Still, we believe this is a viable alternative in the absence of the cover if both hands are allowed.

***Gyro sensor*:** A gyro sensor is another feasible solution without any additional sensors. However, when the sensor is used to control the modes, the noise from one's unintended tilt can affect the accuracy of such interaction (e.g., controlling a device while walking).

Swiping gestures: This might conflict with existing gestures used in some applications. For instance, horizontal bezel swipe is used to go back/forward in a web browser (iOS) and vertical bezel swipe is used for pulling the notification bar (Android, iOS) and control center (iOS). Vertical/horizontal swipes are also commonly used to scroll the contents.

***Extended notification bar*:** As a purely software solution, the content of the notification bar can be extended to show full messages. Although it would be a natural extension to the current interface, it require multiple steps for both task switching and peek: (1) change of grip, (2) swipe from the top, and then (3) select the message. On the other hand, Peek-a-view has a streamlined gesture: lifting the cover slightly with one's thumb and then flipping or releasing the cover to switch or to complete the peek, respectively.

6. CONCLUSION

In this paper, we presented a novel flip cover interaction concept for smartphones, which uses angle between flip cover or sub display and main display. It is designed to enable a range of reconfigurable uses and input modalities. Users can use our system to assist input control or change application modes in a natural manner without occluding the main screen.

We described and evaluated Peek-a-View, a flip-cover-based task switching interface as one example use case scenario of cover angle input system. The proposed method outperformed the conventional button-based approach in terms of task completion time. It was also preferred by the participants based on subjective responses. Peek-a-View functions easily without disturbing standard features and is directly applicable to current devices.

We provided insights about the input methods for devices having secondary display or accessary which can be foldable. We expect foldable di splay such as foldable phone can be on market in the near future. We currently verify only one example input method, Peek-a-View. For future work, we need to explorer more scenarios which can get advantage using cover angle input system and improve the idea. Also we plan to conduct quantitative and qualitative evaluation to define meaningful input values instead using whole angles. Angles from 0^0~360^0 need to be mapped to usable values without errors. They can be normalized to proper values by further user study.

REFERENCES

[1] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage. In *Proceedings of the conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '11), 47-56.

[2] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: multi-"touch" interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (UIST '08), 201-204. http://dx.doi.org/10.1145/1449715.1449746

[3] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1499-1508. http://dx.doi.org/10.1145/2702123.2702425

[4] Jessica Cauchard, Markus Löchtefeld, Mike Fraser, Antonio Krüger, and Sriram Subramanian. 2012. m+pSpaces: virtual workspaces in the spatially-aware mobile environment. In *Proceedings of the conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '12). http://dx.doi.org/10.1145/2371574.2371601

[5] Ayumi Fukuchi, Koji Tsukada, and Itiro Siio. 2013. AwareCover: Interactive

cover of the smartphone for awareness sharing. In *Universal Access in Human-Computer Interaction: Applications and Services for Quality of Life*, Constantine Stephanidis and Margherita Antona (eds.). Springer Berlin Heidelberg, New York, 620–625. http://dx.doi.org/10.1007/978-3-642-39194-1_71

[6] Gizmag. Prototype mobile phone covers foreshadow new wave of E Ink displays, Retrieved August 6, 2015 from http://www.gizmag.com/e-ink-mobile-phone-covers/29063/

[7] Sungjae Hwang, Andrea Bianchi, Myungwook Ahn, and Kwangyun Wohn. 2013. MagPen: magnetically driven pen interactions on and around conventional smartphones. In *Proceedings of the conference on Human-computer interaction with mobile devices and services* (MobileHCI '13). http://doi.org/10.1145/2493190.2493194

[8] Inkcase. Retrieved August 3, 2016 from http://www.inkcase.com/en_product.php

[9] Jaemin Jo, Bohyoung Kim, and Jinwook Seo. 2015. EyeBookmark: Assisting recovery from interruption during reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 2963–2966. http://doi.org/10.1145/2702123.2702340

[10] Yuna Kang, Soonhung Han. 2012. Improvement of smartphone interface using an AR marker. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications*

in Industry (VRCAI '12), 13-16. http://dx.doi.org/10.1145/2407516.2407519

[11] H Ketabdar, KA Yuksel, A JahanBekam, M Roshandel and D Skripko. 2010. Magisign: User identification/authentication based on 3d around device magnetic signatures. In *Proceedings of 4th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (UBICOMM '10), 31-34.

[12] Hamed Ketabdar , Hengwei Chang, Peyman Moghadam, Mehran Roshandel, and Babak Naderi. 2012. MagiGuitar: a guitar that is played in air! In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion* (MobileHCI '12), 181-184. http://doi.org/10.1145/2371664.2371704

[13] Sunjun Kim, Jihyun Yu, Geehyuk Lee. 2012. Interaction techniques for unreachable objects on the touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (OzCHI '12), 295-298. http://dx.doi.org/10.1145/2414536.2414585

[14] Philip Kortum and Mary Sorber. 2015. Measuring the usability of mobile applications for phones and tablets. *International Journal of Human-Computer Interaction* 31, 8: 518-529. http://doi.org/10.1080/10447318.2015.1064658

[15] Luis Leiva, Matthias Böhmer, Sven Gehring, and Antonio Krüger.2012. Back to the app: the costs of mobile application interruptions. In *Proceedings of the conference on Human-computer interaction with mobile devices and services*

(MobileHCI '12). http://dx.doi.org/10.1145/2371574.2371617

[16] Nielsen Norman Group. Multitasking on Mobile Devices, Retrieved August 5, 2015 from http://www.nngroup.com/articles/multitasking-mobile

[17] Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An in-situ study of mobile phone notifications. In *Proceedings of the conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '14), 233-242. http://dx.doi.org/10.1145/2628363.2628364

[18] PopSLATE Media, Inc. 2015. Retrieved August 6, 2016 from http://www.popslate.com.

[19] Christian Rendl, David Kim, Sean Fanello, Patrick Parzer, Christoph Rhemann, Jonathan Taylor, Martin Zirkl, Gregor Scheipl, Thomas Rothländer, Michael Haller, and Shahram Izadi. 2014. FlexSense: A transparent self-sensing deformable surface. In *Proceedings of the symposium on User interface software and technology* (UIST '14), 129–138. http://doi.org/10.1145/2642918.2647405

[20] Christian Rendl, David Kim, Patrick Parzer, Sean Fanello, Martin Zirkl, Gregor Scheipl, Michael Haller, and Shahram Izadi. 2016. FlexCase: Enhancing Mobile Interaction with a Flexible Sensing and Display Cover. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 5138-5150. http://dx.doi.org/10.1145/2858036.2858314

[21] Meredith Ringel. 2003. When one isn't enough. In *Proceedings of the*

SIGCHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI '03), 762-763. http://doi.org/10.1145/765891.765976

[22] Fletcher R. Rothkopf. 2012. Cover attachment with display, Patent No. US20120194448, Filed August 11, 2011, issued August 2, 2012.

[23] Martin Spindler, Marcel Martsch, and Raimund Dachselt. 2012. Going Beyond the Surface: Studying Multi-Layer Interaction Above the Tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 1277-1286. http://dx.doi.org/10.1145/2207676.2208583

[24] Martin Spindler, Wolfgang Büschel, and Raimund Dachselt. 2012. Use Your Head: Tangible Windows for 3D Information Spaces in a Tabletop Environment. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces* (ITS '12), 245-254. http://dx.doi.org/10.1145/2396636.2396674

[25] Statista. Most used instant messaging apps 2015, Retrieved September 10, 2015 from http://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/

[26] Voice Actions, Retrieved August 10, 2016 from https://developers.google.com/voice-actions/

[27] Sang Ho Yoon, Ke Huo, and Karthik Ramani. 2016. TMotion: Embedded 3D Mobile Input using Magnetic Sensing Technique. In *Proceedings of tenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '16), 21-29. http://dx.doi.org/10.1145/2839462.2839463

[28] Martina Ziefle. 1998. Effects of display resolution on visual performance. *The Journal of the Human Factors and Ergonomics Society* 40, 4: 554-568. http://dx.doi.org/10.1518/001872098779649355

국문 초록

스마트폰 커버를 이용한 모바일 인터렉션

서울대학교 대학원

컴퓨터공학부

최고은

대부분의 스마트폰은 사용자가 여러 어플리케이션을 동시에 사용할 수 있도록 멀티 테스킹을 지원하고 있다. “Recent apps”나 “Back” 버튼과 같은 S/W, H/W 버튼을 제공하여 어플리케이션을 변경할 수 있게 해주거나, 화면을 나누어 여러 가지 어플리케이션을 동시에 볼 수 있도록 하고 있다. 하지만, 작은 화면에서 동시에 여러 어플리케이션을 동작시키는 것은 낮은 사용성을 보이며, 현재 테스크를 방해하는 알림 메시지에 응답하는 등의 예와 같이, 어플리케이션 간의 전환이 빈번할 경우, 현재의 지원 방식은 처리하기가 매우 번거롭다. 본 연구는 서브 테스크를 위해 추가적인 가상의 스크린 스페이스를 제공함으로써 테스크 전환 오버헤드를

줄일 수 있도록 완전히 투명한 플립핑 스크린 커버로 만든 Peek-a-View를 이용하는 것을 제안한다. 이를 위해 알림이 왔을 때, 핸들링하는 인터페이스 동작을 구현하여 Peek-a-View의 feasibility를 측정하였다. 알림이 왔을 때, 사용자는 앱을 전환할 필요 없이, 커버를 살짝 들어 알림의 컨텐츠를 확인할 수 있고, 테스크 전환이 필요하다면, 사용자는 완전히 커버를 열어 알림을 발생시킨 해당 앱으로 전환하여 작업할 수 있다. 두 개의 사용자 실험은 flipping과 peeking 인터렉션이 서브테스스크와 연관되어 있는 작업들에 대해 유용하며, 향상된 성능을 제공함을 보여준다.

스마트폰 커버 인터렉션은 멀티 테스킹 시나리오 외에도 추후 연구에서 확장될 수 있다. 스마트폰 커버의 각도에 따라 비연속적인 플립 커버의 상태를 알려주는 이벤트와 플립 커버를 움직일 때 연속적으로 값의 변화를 알려주는 각도의 움직임 이벤트로 나누어 지원할 수 있도록 두가지 타입에 대한 input 모달리티를 제안한다.

키워드 : 커버 인터렉션, 모바일 기기, 테스크 전환

학번 : 2015-21268