# Investigating Cross-Device Interaction Techniques: A Case of Card Playing on Handhelds and Tablets

**Mikael B. Skov, Jesper Kjeldskov, Jeni Paay, Heidi P. Jensen, Marius P. Olsen**
Centre for Socio-Interactive Design, Aalborg University
Selma Lagerlöfs Vej 300, DK-9220 Aalborg East, Denmark
{dubois, jesper, jeni}@cs.aau.dk, {hsn.mobil, marius.pallisgaard}@gmail.com

## ABSTRACT
Cross-device interaction is getting more and more common as mobile and handheld technologies surround us in almost every situation and context. HCI research has started to study various aspects of cross-device interaction including application areas and techniques, but we still need further studies on opportunities. We illustrate six cross-device interaction techniques. Using two different prototype systems that implement them for a card-playing context we evaluate them in two studies on usability and usefulness. Our findings suggest that some techniques were fast to perform (*touch-and-hold* and *tap-tap*) and some had a high number of errors (*swipe*). Also, our participants would sometimes mix up the techniques and make a wrong action while playing the game. Finally, we discuss and illustrate issues of cross-device interaction consistency and division of shared and personal devices.

### Author Keywords
Cross-device interaction, handheld devices, card games

### ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

## INTRODUCTION
In recent years we have witnessed a growing interest in mobile device interaction involving more than one device. Often referred to as cross-device interaction or multi-device interaction, we now see situations where people have access to more than one device, either when they are alone (i.e. use a smartphone and a tablet) or together with other people (where people use their own phones). For instance, we often see groups of people in social situations where they use smartphones as part of the social interaction in the group, e.g. sharing photos (Nielsen et al., 2014). It becomes increasingly important to be able to design for cross-device interaction.

Within ubiquitous computing research, the idea of using numerous devices has been considered important for many years. Rekimoto (1998), for example, envisioned almost two decades ago what he called multiple-computer user interfaces and argued that interaction techniques must overcome the boundaries among devices in multi-device settings. HCI research has recently begun to study opportunities of user interaction across several platforms (e.g. Boring et al., 2009; Döring et al., 2010, Marquardt et al., 2012; Ohta and Tanaka, 2012), e.g. interaction techniques for sharing documents between collocated people (Marquardt et al., 2012) and cross-device interaction using wearable devices (Chi and Li, 2015) or Smartwatches (Houben and Marquardt, 2015). But we still have a limited understanding of how we can interact with more devices at the same time and how such interaction techniques could be applied.

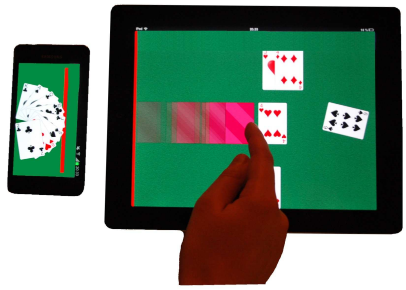


**Figure 1. Cross-device interaction in our card game, illustrating tablet-to-phone interaction where a user moves a card from the tablet to the phone with a swipe gesture**

Inspired by such challenges, this paper reports from two studies on cross-device interaction for games. Based on a workshop, we illustrate six different interaction techniques that can be used for handheld-to-tablet interaction or tablet-to-handheld interaction. We evaluate the usability and usefulness of these techniques for playing cards on several devices in two different studies. Our findings show that the six techniques differed in performance, e.g. the interaction technique imitating the gesture players perform when they play a card, *wrist-whip,* was slowest of the handheld-to-tablet interaction techniques, but it was also the one with fewest interaction errors. Also, people would sometimes mix up the techniques and thus perform a wrong action in the system. Finally, we discuss issues related to cross-device consistency and division devices.

## RELATED WORK
The number of smartphones and tablets has increased significantly over the last 5 years, and it is now not uncommon for people to have several of these devices in use. This creates new opportunities for exploring apps and interaction designs that span across multiple devices rather than being limited to one. Researchers have responded to these opportunities by exploring cross-device applications and interactions that allow users to



OzCHI '15, December 07 - 10 2015, Melbourne, VIC, Australia

DOI: http://dx.doi.org/10.1145/2838739.2838763

link applications across individual devices and interact with them as one. Research investigating new interaction techniques for cross-device interaction include *Pass-Them-Around* by Lucero et al. (2011), *GroupTogether* by Marquardt et al. (2012), *Duet* by Chen et al. (2014), *JuxtaPinch* by Nielsen et al. (2014), *Smarties* by Chapuis et al. (2014), HyPR by Houben et al. (2014), and *Conductor* by Hamilton and Wigdor (2014).

In an early study of cross-device interaction, Alsos and Svanæs (2006) studied techniques for using handheld devices together with stationary displays viewing x-ray images in a hospital setting. Their study found that the best interaction techniques in this scenario were those that resided on the handheld device, using the stationary display as a receiver for media content. Furthermore the study illustrated that cross-device interaction techniques are affected by ergonomic and social factors of their use context. Focusing also on interaction across handheld and wall-mounted displays, Boring et al. (2009) explored how mobile phones can be used for controlling a pointer on a larger public display using the three techniques of scroll, tilt, and move. The study showed that while techniques using motion resulted in faster task completion time, the novelty of this type of interaction also resulted in high error rates. In a related study Döring et al. (2010) found that introducing different levels of privacy into a cross-device application, i.e. private information spaces on handhelds and public information spaces on tabletops, creates some powerful design opportunities, exemplified with the case of a card game.

In *Pass-Them-Around,* Lucero et al. (2011) explore cross-device interaction in collaborative photo sharing on mobile phones. They present several interaction techniques based on conventional photo sharing practices, and evaluate them in use on mobile devices. The study shows that cross-device interaction worked well for the specific case, but also indicates that the interaction techniques for this could benefit from being designed in ways that would make them work beyond the table, for example, while people are holding the device in their hand. Also focusing on mobile devices, in *GroupTogether,* Marquardt et al. (2012) study cross-device interaction on tablets for co-present collaboration. Based on frameworks of F-formations and micro-mobility, their prototype supports fluid and minimally disruptive interaction in document transfer by leveraging the proxemics of people to devices. Introducing smart watches into the design space for cross-device interaction, *Duet* by Chen et al. (2014) coordinates motion and touch-input across a phone and a watch, and makes their visual and tactile output capabilities available to one another. Thereby the watch is transformed into an active element of the device unity enabling multi-device gestures and multi-device sensing techniques. In a similar manner *JuxtaPinch* by Nielsen et al. (2014) allows a number of mobile devices to be put next to each other and "pinched" together to form a larger collaborative workspace with interaction going across several devices. In this study it was found that creating a larger display workspace by placing multiple mobile devices next to each other facilitated different social interactions than individual devices alone. It was, however, found that cross-device interaction resulted in some challenges of synchronization when multiple users were interacting simultaneously.

Including wall-size displays into the cross-device ecology, *Smarties* by Chapuis et al. (2014) is an input system for collaborative wall displays that integrates mobile devices into a cross-device application platform, enabling multiple cursors, keyboards, widgets and clipboards. Rather than experimenting with specific interaction techniques, this work focuses on facilitating real-world implementation of cross-device applications through a concrete infrastructure, communication protocol, and library. A similar focus on infrastructure for real-world cross-device applications can be found in *HyPR* (Hybrid Patient Record) by Houben et al. (2014) who present a mobile device that merges paper and electronic patient records and facilitates interaction across this device and shared displays in hospital wards.

Aggregating much of the previous work on cross-device interaction, *Conductor* by Hamilton and Wigdor (2014) is a prototype framework for cross-device applications and interaction, combining a number of interaction techniques for sharing information, chaining tasks, and managing interaction session across multiple mobile devices. In the work by Nebeling et al. (2014), a similar high-level perspective is taken in the presentation of a new GUI builder for supporting the development of cross-device web interfaces, allowing designers to simulate multiple devices on their application development platform, and allowing the implementation of cross-device applications to take place across multiple devices.

## CROSS-DEVICE INTERACTION TECHNIQUES

This section illustrates six interaction techniques for cross-device interaction on mobile phones and tablets for the purpose of playing cards. We chose card playing as our case for cross-device interaction as it typically involves multiple users but also private as well as shared interaction spaces (the individual player's hand of cards, and the cards on the table). Also, we are currently seeing more commercial card game products for tablets and smartphones coming out.

In order to generate ideas for different cross-device interaction techniques, we held a workshop with nine graduate students and two university professors working within the area of HCI. The workshop comprised three activities: brainstorming, refinement, and evaluation. We provided participants with materials for exploring and illustrating cross device interaction for card playing, i.e. playing cards and mobile devices. The workshop resulted in 44 interaction techniques for nine card game interactions. From these we selected six notably different techniques to explore and compare empirically – three for playing a card from one's hand (*swipe*, *touch-and-hold*, and *wrist-whip*) and three for drawing a card from the table (*directional-swipe*, *drag*, and *tap-tap*).

In the following, we describe these six interaction techniques in detail. In the following two sections, we describe how we evaluated their usability and usefulness for cross-device interaction through two experiments.

### Handheld-to-Tablet Interaction: Playing a Card

We selected three interaction techniques for playing a card from one's hand to the table: *swipe, touch-and-hold,* and *wrist-whip* (illustrated in figure 2). These techniques enable the user to "play a card" from a handheld device (mobile phone) transferring it to the shared device on the table (tablet). In the following we explain the three techniques using playing cards as the case example.

**Figure 2. The three cross-device interaction techniques for playing a card: *swipe*, *touch-and-hold*, and *wrist-whip*.**

The first interaction technique is *swipe.* This is a common and well-known gesture for touchscreen interaction often used, for example, for shifting between pages on a tablet or mobile phone, or for scrolling a page of content that exceeds the size of the screen. It is done by moving one's finger on the screen surface in a linear motion. As illustrated with a blue dot and arrow in figure 2 (left), our use of the swiping technique requires the user to touch the card to be played, and swiping it towards the edge of the mobile phone. This transfers the card to the tablet.

The second interaction technique is *touch-and-hold.* This is a less common technique for touchscreen interaction, but has the advantage of requiring less physical motion. In our implementation the *touch-and-hold* interaction is performed on the card the user wants to play, as illustrated in figure 2 (center). When touching a card and holding the touch for a short period of time (500ms), that card is then transferred to the tablet.

The third interaction technique is *wrist-whip.* This is an uncommon technique, imitating the motion a card player makes when flicking a card onto the table. In our implementation the user selects the card to be played by touching it on the screen with their thumb and then, while holding the card, performing a whip-like movement with the wrist to mimic a flick. Apart from the touchscreen, this technique uses motion sensors in the phone to register wrist-movements.

### Tablet-to-Handheld Interaction: Drawing a Card

Complementing the three Handheld-to-Tablet techniques, we selected three interaction techniques for drawing a card: *directional-swipe, drag,* and *tap-tap* (figure 3). These techniques enable the user to "draw a card" from the table to their hand, transferring it from the tablet to their handheld device.

The first technique for drawing a card is *directional-swipe*. This is very similar to the *swipe* technique for playing a card in that it requires the user to touch a card to be drawn from the deck, and then swiping it toward the edge of the tablet. In contrast to the *swipe* technique for playing a card to the table, when drawing a card the user must swipe it in a direction from the deck toward their physical placement around the shared device.

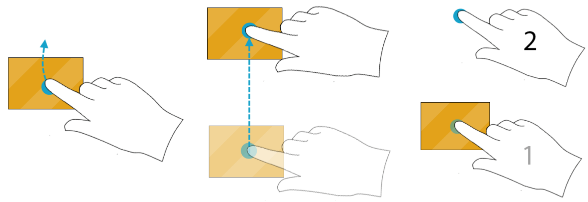



**Figure 3. The three cross-device interaction techniques for drawing a card: *directional-swipe*, *drag*, and *tap-tap*.**

The second technique for drawing a card is *drag.* This is a well-known technique within direct manipulation interfaces described by, for example, Dix et al. (2003) and Shneiderman (1998) and is also common in touchscreen interaction. In our implementation of *drag*, the user continuously touches and holds the card while sliding their finger toward the edge of the shared device closest to them, effectively "dragging" the card. When the card touches the edge of the tablet's screen it is then transferred to the player's handheld device.

The third draw card technique is *tap-tap*. This is a part of a common catalogue of tapping techniques often used for touchscreen interaction (Microsoft, 2015). In our implementation of the technique, the user has to first tap the deck of cards to draw from it, and then tap the edge of the tablet screen nearest a player to indicate which one the card should be dealt to.

## STUDY A: USABILITY

The six cross-device interaction techniques described above were implemented in a prototype system and evaluated in a study investigating their overall usability.

### Participants

Eighteen people participated in the experiment, 13 males and 5 females. The participants were between 22 and 37 years old (M=27.1, SD=4.1). Sixteen of them were right-handed and 14 owned their own smartphone. Eight of the participants were computer science or software engineer students, three were students from other faculties, and 7 were industry software developers. The participants were recruited through our social network.

### Experimental Design

We applied a within-subject research design with the six interaction techniques as independent variables and different task completion times and interaction errors as dependent variables. We balanced the order of techniques for the 18 subjects to minimize learning effects using a Balanced Latin Square Design.

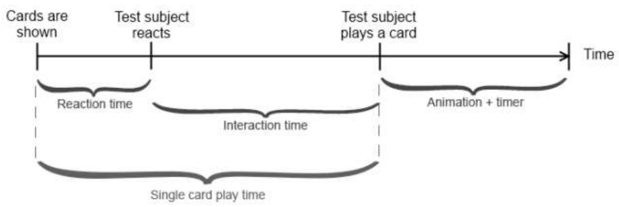



**Figure 4. The dependent variables in our study namely single card play time, reaction time, interaction time, and total task completion time**

We measured these completion times when playing or drawing a card (illustrated in figure 4): single card play time, reaction time, interaction time, and task completion time (total). *Single card play time* refers to when the time it takes from the cards being shown on the screen, to when the test subject plays or draws a card (a card is played/drawn when the system has recognized that the current interaction technique has been performed on a card). This time does not include the animation and timer (1 second for the animation + 200 milliseconds for the timer).

*Reaction time* is the time from the cards being shown on the screen to the first time the user touches the screen, i.e. the time it takes for the test subject to react. This is thought to be the time it takes for the test subject to recognize the lowest ranked card in the hand (in the play card case) or the target player (in the draw card case).

*Interaction time* is the time from the test subject touching the screen for the first time until they have performed the interaction technique and it is recognized by the system, i.e. the time used to play/draw a card. This can reflect the difficulty of performing the interaction technique.

*Total task completion time* is the time it takes to complete a task. The task consisted of playing or drawing a card 30 times. Hence, the total task completion time is the sum of 30 individual single card play times.

Furthermore, we measured interaction errors and playing errors made during the evaluation. *Interaction errors* refer to the number of gestures made before the expected interaction technique was successfully carried out, which indicates how difficult it is to perform a specific interaction technique. *Playing errors* refer to the number of times a test subject plays a wrong card from their hand or draws a card to the wrong player.

## Implementation

We implemented the six interaction techniques in two parts with one part for playing a card from one's hand and the other for drawing a card from the table. We refer to them as the *play card* and the *draw card systems*. The play card system was developed to test handheld-to-tablet interaction. The draw card system was developed to test tablet-to-handheld interaction.

Five interaction techniques (*swipe*, *direction-swipe*, *touch-and-hold*, *tap-tap* and *drag*) were implemented using the JavaScript library Hammer.js using the default values provided by Hammer.js. The last interaction technique (*wrist-whip*) was implemented using the accelerometer of the smartphone, through the HTML5 device orientation API (LePage, 2015). *Wrist-whip* is recognized by the system when the front-to-back tilt is greater than 40 degrees, and the z-axis rotation is greater than 140 degrees. For the experiment, the two prototype systems ran on a tablet (Acer A200) and on a smartphone (Samsung Galaxy S2).

### *The Play Card System*

We developed the *play card system* to evaluate the three interaction techniques for playing a card (*swipe, touch-and-hold*, and *wrist-whip*). The basic idea of the prototype is that the user is presented with a hand of cards (seven cards as illustrated in figure 5). The *play card system* works in the following way. The system chooses seven randomly picked cards and arranges them randomly on the screen. The user must then play the lowest ranked card in the hand as fast as possible with as few errors as possible using one of the three interaction techniques.

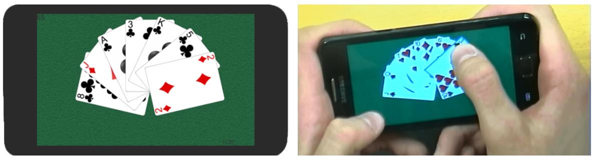


**Figure 5. The play card system where seven randomly picked cards are shown to the user**

The experiment consisted of three sessions with one session for each technique. A session consisted of a learning part and the actual test part. Before starting the test part, the system allows the user to practice the interaction technique. Here the user will be presented with a hand of seven cards from a regular deck of cards and the cards are chosen and placed randomly. The test subject must then play five cards with the current interaction technique. The test subject must play the lowest ranked card available (otherwise a play card error will be registered) where the Ace is the lowest ranked card and King is the highest ranked. The actual test followed the same procedure but consisted of playing a card 30 times. For every subject, the system automatically controlled the events and logged the interaction.

### *The Draw Card System*

The *draw card system* was developed to test the three interaction techniques for drawing a card (*directional-swipe, drag*, and *tap-tap*). The test subject is presented with a squared tablet view representing a playing table. For each side, there is a marking that reflects a player at the table. In the middle is a deck of cards where the test subject must deal cards from the deck to one of the four players on the table (see figure 6). The user has to give a card to the player (table edge), which is colored yellow. The player who is to receive a card is randomly selected. The distance from the middle of the deck to the border of a player is equal for all players to ensure consistent play times. Again, the test consists of a learning part and a real test part (as with the *play card system*). After the learning part, in the actual test the user must deal 30 cards.

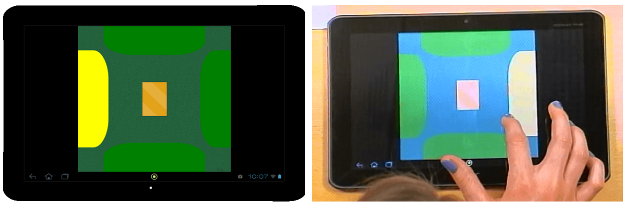


**Figure 6. The draw card system where the user has to draw cards to one of four players (the yellow edge)**

## Procedure

The experiment was conducted in the usability laboratory at the Computer Science Department, Aalborg University. First, each participant was informed about purpose of the experiment – to test the usability of six interaction techniques, and was informed that the session was being videotaped. We then collected demographic information. The system was then set up to present the test subject with the interaction techniques in the order required by the participant's position in the Latin Square Design. Secondly, we began the test session. The subject received the device with the first interaction technique and the test monitor explained the interaction technique. The subject then played 5 learning plays followed by 30 real plays. This was repeated for the three interaction techniques. Most test sessions took around 15 minutes.

### Data Analysis

All of the dependent variables were logged automatically by the test system and saved into a database. However, the interaction errors for the *wrist-whip* were partially logged manually. All the interaction errors when interacting with the screen were logged by the test system, but the whip motions could not be logged automatically, and were thus counted manually by observation. The results were analyzed using one-way repeated-measures ANOVA for 3 correlated samples (Lowry, 2001). If an ANOVA showed a significant difference between the three interaction techniques, we used Tukey HSD Test for Post-ANOVA Pair-Wise Comparisons in a One-Way ANOVA to determine which of the interaction techniques the significant difference was between.

## RESULTS

In the following the findings of our experiment will be presented. First the task completion time results will be presented followed by the task error results.

### Task Completion Times

The interaction technique efficiency was expressed by total task completion time. Each interaction technique was performed 30 times, and table 1 shows that *touch-and-hold* was faster than the two others. It took the participants 71.59 seconds to complete the 30 *touch-and-hold* interactions, *swipe* took 81.81 seconds, while *wrist-whip* took 84.94 seconds. This difference is significant according to a one-way repeated-measures ANOVA $F(2,34)=4.34$, $p<0.021$. A pair-wise post hoc comparison Tukey HSD test showed that the difference in total task completion time was significant between *touch-and-hold* and *wrist-whip* ($p<0.05$). However, the difference can be partly explained by the fact that *wrist-whip* is a two-step interaction technique where the card is first selected and then the whip motion is performed. We further identified that some participants had problems when performing the *swipe* technique. We will return to this later.

| | Touch & Hold (n=18) | Wrist-Whip (n=18) | Swipe (n=18) |
|---|---|---|---|
| Single Card | 2.39 (0.42) | 2.83 (0.60) | 2.73 (0.64) |
| Reaction | 1.85 (0.43) | 1.94 (0.55) | 1.86 (0.35) |
| Interaction | 0.54 (0.05) | 0.89 (0.20) | 0.87 (0.56) |
| Total (30 cards) | 71.59 (12.67) | 84.94 (17.94) | 81.81 (19.18) |

**Table 1. Results from the *play card system* usability test**

We refer to each play card interaction as single card play and is defined as beginning when cards are shown on the display and ending when a card is played. Unsurprisingly, participants spent less time performing the *touch-and-hold* technique than the two other techniques. Participants using *touch-and-hold* show 12.23% less time usage than *swipe* and 18.41% less than *wrist-whip*, and this difference is significant $F(2,34)=4.32$, $p<0.021$. A post hoc test shows that significant difference lies between the *touch-and-hold* and the *wrist-whip* techniques ($p<0.05$).

Within each play action, we logged the user's reaction time, expressed in seconds, from the time the cards were shown to the participant first interacting with the screen. As table 1 shows we found little difference between the three techniques for reaction time. Thus, it seems that interaction techniques had no significant effect on how fast the participant would react. On the other hand, we found a significant difference in the interaction time. Interaction time is defined as the time between the player reacting and the card being played. We did not directly log this time, but calculated it by subtracting the reaction time from the single card play time. Table 1 shows a significant difference between both the *touch-and-hold* and *wrist-whip* 64.82%, and the *touch-and-hold* and *swipe* 61.11%. $F(2,34)=5.68$, $p<0.007$.

In the draw card condition we compared *directional-swipe*, *drag*, and *tap-tap*. We logged total task completion time on the tablet in the same manner as total task completion time for the play card condition. Each interaction technique was also performed 30 times. As table 2 shows *drag* was slower than the other two interaction techniques taking in total 39.38 seconds against *direction-swipe* 33.86 and *tap-tap* 32.87. An ANOVA repeated-measures test shows that this difference is significant $F(2,34)=6.0$, $p<0.006$ and a Tukey post hoc test shows a significant difference between both *drag* and *direction-swipe* ($p<0.05$) but also *drag* and *tap-tap* ($p<0.01$).

| | D-Swipe (n=17) | Tap-tap (n=17) | Drag (n=17) |
|---|---|---|---|
| Single Card | 1.13 (0.32) | 1.10 (0.15) | 1.31 (0.26) |
| Reaction | 0.79 (0.15) | 0.72 (0.12) | 0.79 (0.21) |
| Interaction | 0.33 (0.22) | 0.37 (0.11) | 0.53 (0.14) |
| Total (30 cards) | 33.86 (9.47) | 32.87 (4.45) | 39.38 (7.92) |

**Table 2. Results from the *draw card system* usability test**

The single card draw completion times show that *drag* is significantly slower than the two other techniques $F(2,34)=6.1$, $p<0.006$. Again the post hoc test shows that the significant difference occurs between *drag* and *directional-swipe* ($p<0.05$) and *drag* and *tap-tap* ($p<0.01$). A technical error meant that some data was not recorded for one of the participants which was removed from the three interaction techniques in the statistical analysis of both reaction time and interaction time (as table 2 shows where n=17). But as with play card, we found no significant differences for reaction time for the techniques with draw card. The interaction times for *directional-swipe*, *tap-tap* and *drag* are also illustrated in table 2 where we see that *directional-swipe* is significantly faster than both *tap-tap* 12.12% and *drag*

60.60% $F(2,32)=8.14$, $p<0.001$ and is confirmed by a post hoc test. Furthermore, we observe a tendency here that *directional-swipe* is generally faster than the other techniques, but also that the standard deviation is high indicating that they have rather different completion times. This tendency was also seen for the *swipe* technique for play card. This is not surprising as it is the same implementation of the actual swipe motion.

### Task Errors

Besides task completion times, we further measured errors as interaction errors and playing errors. Our results show that our participants using *swipe* to play cards made many more errors (on average 17.94) than the two other techniques (1.39 and 1.17) and this difference is not surprisingly significant $F(2,34)=22.63$, $p<0.001$. With an average of 17.94 errors for 30 play cards, the *swipe* has 1190.65% more errors than *touch-and-hold*, which a post hoc test confirms is significant ($p<0.01$), and 1433.33% more errors then the *wrist-whip* which according to the post hoc test is also significant ($p<0.01$). We clearly found that *swipe* was the interaction technique which users have the most difficulty performing when playing cards. However we see from *swipe* standard deviations in table 3 that participants performed quite differently.

| | Touch & Hold (n=18) | Wrist-Whip (n=18) | Swipe (n=18) |
|---|---|---|---|
| Int. errors | 1.39 (1.58) | 1.17 (1.29) | 17.94 (14.70) |
| Play errors | 1.11 (1.32) | 1.39 (2.25) | 1.44 (1.50) |

**Table 3. Results from the usability test of the *play card system* on interaction and play errors.**

Playing errors are defined as when the test subject plays the wrong card from his hand. The figures seen in table 3 are the sum of all the playing errors from 30 interactions. Even though we found several interaction errors for *swipes*, this did not affect playing errors where we found no significant differences between the three techniques. In fact, they are quite close on average numbers, but we also see that they have high standard deviations indicating that participants performed quite differently.

| | D-Swipe (n=17) | Tap-tap (n=17) | Drag (n=17) |
|---|---|---|---|
| Int. errors | 4.94 (4.43) | 0.61 (1.46) | 1.28 (1.64) |

**Table 4. Results from the usability test of the *draw card system* on interaction and play errors.**

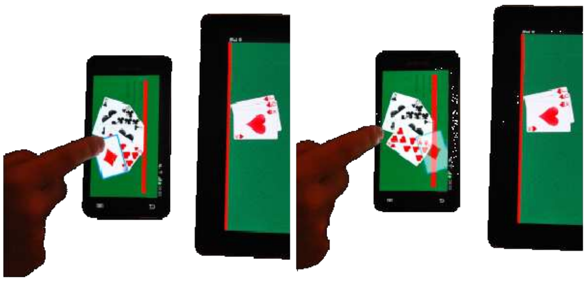
When we consider the draw card condition results, we see that *directional-swipe* has an interaction error rate of 4.94 errors per 30 interactions, which was much higher than *tap-tap* (0.61 errors) and *drag* (1.28 errors). This was significant according to an ANOVA test $F(2,32)=11.28$, $p<0.001$ and a post hoc shows that the difference is in fact significant between *directional-swipe* and *tap-tap* ($p<0.01$) and *directional-swipe* and *drag* ($p<0.01$).

## STUDY B: USEFULNESS

The second study in this paper explores the usefulness of cross-device interaction for card games. We chose *touch-and-hold* as the interaction technique for the mobile device (the card hand), as it performed best overall in the usability study, and both *drag* and *directional-swipe* as interaction techniques for the tablet (the table) in this study. Even though the *tap-tap* technique worked quite well in the first study (completion times and errors), we chose the two other techniques to complement the *touch-and-hold* technique and to explore the slightly more complicated interactions in *directional-swipe* and *drag*.

### Card Playing Prototype

We implemented a card playing prototype system that allows four players to play different kinds of card games. The system supports one tablet and up to four smartphones where the tablet acts as the table space of a card game (the shared space) and the smartphones act as the player's hand in card games (personal and private). The system itself imposes no card playing rules upon the users, players may decide for themselves which card game to play. The game system is an open world system and it is therefore the responsibility of the players to maintain the rules. They can play and draw cards as they like. When the game starts the tablet shows a deck of cards and the smartphones are empty because no cards have been dealt yet. Color bars at the edge of the tablet (and on the smartphone) indicate the positions of each of the players in the game (the red line in figure 7).



(a) Hold initiated. (b) Card being played.

**Figure 7. Playing a card from one mobile phone to the tablet using touch-and-hold as interaction technique**

The player's user interface shows a hand of cards as in the usability test. In addition, there is a color bar on top of the screen to indicate the player's position around the table. To play a card onto the table (tablet), the players must use the *touch-and-hold* interaction technique. Once a card has been played, it will animate off of the screen as in the usability test system (as illustrated in figure 7).

On the tablet, an edge has the same color as the player's color bar. The tablet only shows color-borders for the number of players that have joined the game (maximum four). A player must *swipe* the deck in the direction of his own color to deal a card to himself. The card will fly to the border of the corresponding player once a card has been drawn. This applies to both deck and player pile cards. *Drag* is used to move cards from one position to another on the table. Clicking on a card in a stack will put that card on top of the stack. For practical reasons, we also implemented a gesture for rotating cards.

### Participants and Procedure

Eight people in three groups participated in the usefulness study including two groups with three participants, and one group with two participants. We recruited them among friends and family and represented both regular and non-regular card players. In the following, we will elaborate on the participants, which card games they played, and their use of our card playing system.

Group A consisted of three participants between 50 and 55, they were regular card players and all of them owned a smartphone. The test was performed with an iPad as the table, a Samsung Galaxy S2, a Huawei Ascend P1, and an iPhone 4 as the three player hands. The participants played two different card games using our system. First, they played a card game called “President and the Bum”. Here, each player is dealt ten cards and the goal is to get rid of the cards in your hand. The player who does so first, wins. Besides dealing cards from the deck (on the tablet), the primary interaction takes place on the smartphone. The second game they played was “Go Fish”, where each player starts with seven cards and has to get as many pairs as possible by asking other players for cards. Here most interaction is performed on the table. When a player gets a card from another player, the card is played to the table and picked up again.

Group B consisted of two participants aged 24 and 27. They would only play cards occasionally and they both owned a smartphone (Samsung Galaxy S3 and Samsung Galaxy S2), which were used in the study. An iPad acted as the table in their case. They played a card game called “Zero”, where each player is dealt seven cards. Players take turns to draw one card and then play one card, i.e. they exchange cards. When a player thinks she has the lowest hand she may knock on the table and after a last round is played, the players must show their hands to identify the winner. The winner gets one card less in their hand for the following rounds. This continues until a player has zero cards in their hand and wins the game.

Group C consisted of three participants all aged 25. They rarely played cards and all owned a smartphone. The test was performed with an iPad (acting as the table), two Samsung Galaxy S2, and one Samsung Galaxy S3. They also played “President and the Bum” like group A (for explanation of the card game see group A). Figure 8 illustrates group C during their card playing activity.

**Figure 8. One group of three participants play a card game using our prototype system.**

We collected results from the study through observations and interviews (held after the card playing sessions). This was done informally, and sessions were held in private situations and in the homes of the participants. We note interactions and comments made during card playing, and interviewed participants on their experiences afterwards.

## RESULTS

Our results generally showed that it was possible for the groups to play different card games on our system. The open world structure of the game where no card rules are enforced or provided, e.g. players can play, draw and arrange cards on the table whenever they feel like it, resulted in every game starting with a discussion among the players to reach agreement of the exact version of the game to be played to ensure that it ran smoothly. In particular, the regular card players (group A) liked the fact that the system provided an open structure for the game, and said “*being able to decide upon the rules made the game more playable*”. They did however argued that because of the limitation of only one deck of cards and not being able to customize cards in the deck, the number of different card games they could play were limited. Interestingly, the non-regular card players (groups B and C) requested rules in a system like this, e.g. functionality for automatically dealing cards to all players. Another interesting observation was that our participants would try to make a correlation between where they were physically seated and their virtual seating in the game. Mostly, they were seated toward the edge at which they had to *directional-swipe* to draw a card. It was observed that they made the swipe towards themselves. Meaning that if they sat at an angle, the gesture would be directed towards them and not perpendicular to the edge. Finally, some participants found it difficult to tell the different cards apart because of the graphics.

As illustrated we used *touch-and-hold* for playing a card, *directional-swipe* for drawing a card, and *drag* for re-arranging cards on the table. The participants quickly learned to use the different interaction techniques, but they would sometimes mix them up (i.e., attempt to use *swipe* to play a card). This could possibly be attributed to learning effects, but could also suggest more fundamental challenges of using several interaction techniques in the same game. Using *directional-swipe* to draw a card (a card which can be manipulated on the table by *drag*) was no problem. Arranging pairs on the table by dragging the cards did not show any difficulties. But in a few cases a *drag* interaction was recognized as a *directional-swipe* and the player receiving the card in their hand would have to play it back to the table.

Several participants made comments on how to improve our system. Most of them requested functionality for rearranging the cards in their hand, e.g. by manually dragging cards left or right or automatic arrangement. One participant said that it would be good if recently played cards were marked or highlighted to emphasize that a card had just been played. Also, this could include mechanisms to emphasize the next player to play a card. A rather interesting limitation of our system was the lack of information on numbers of cards in the hands of opponents. This was clearly important information for the players when playing cards. Thus, it was suggested that the number of cards a player had in his hand (on his mobile device) was showed on the players edge of the tablet, as a picture of the back of the hand of cards held by them or as a number. Another participant added that she would normally interpret how an opponent arranged cards in their hand, so it was suggested that if the cards were shown on the table with the back up, an animation could indicate when people arranged their cards.

## DISCUSSION

Our work investigated cross-device interaction for the case of playing card games. We illustrated six interaction techniques applicable for handheld-to-tablet interaction or tablet-to-handheld interaction. Our results showed that the six techniques differed in their performance, but also shared similarities. For example, the interaction technique imitating the gesture players perform when they play a card *wrist-whip* was the slowest of the three handheld-to-tablet interaction techniques, but it was also the one with fewest interaction errors. We found that the *swipe* gesture would cause significant problems for our participants in that they often made an interaction error while trying to swipe during handheld-to-tablet interaction. On the other hand, we identified only minor differences for the tablet-to-handheld interactions, as they were quite similar, except that the *drag* interaction was slower than the other two. During our second study, we saw that cross-device interaction could work for card playing, as participants were actually able to play different card games on our prototype. While the results from our studies of cross-device interaction constitute a key contribution of our work, we identify a number of additional contributions for cross-device interaction.

### Cross-Device Consistency

We discovered consistency issues or problems in relation to the implemented interaction techniques in the prototypes. In our first prototype, we implemented six techniques namely *swipe, touch-and-hold,* and *wrist-whip* for the handheld-to-tablet interaction and *directional-swipe, drag,* and *tap-tap* for the tablet-to-handheld interaction. While all participants used all six techniques (within-subject design), we found no immediate and significant problems of participants mixing up the techniques during the different rounds of use. Here it seemed that the repetitive nature of the assignment helped participants avoid making those kinds of errors.

However, in our second study (which was done as a field study without clear tasks) we discovered problems related to consistency, as participants would mix up techniques between the different kinds of devices. A key problem for our participants was that the different activities on the table, e.g. to draw/deal a card, or to move a card on the table, are very closely related. In both cases, the user has to move the card either to another place on the table (using *drag*) or to another player (using *directional-swipe*) aiming at the table edge. Participants would mix up these interactions resulting in, for example, a player receiving a card she was not supposed to. These kinds of unintentional interactions are common for user interfaces and designers need to consider this in designing cross-device interaction when similar but different actions require different interactions. While we have no direct evidence for solutions to this problem, our results indicate that the more manual the interaction technique, the fewer mistakes are made in the interaction. Thus, this problem was more often found for *swipe* than for *drag*.

### Division between Shared and Personal Devices

When faced with a situation and context where more people use more devices, designers need to understand differences between information and interaction on shared devices and personal devices. In our case, the division between shared and personal devices was quite clear, as participants would use their mobile phone as a private device (their hand of cards in the game) and the tablet as a shared device (the table and the deck of cards). In fact, when playing a game of cards, the hand is very private (secret) and the mobile device keeps this private from other card players. This clear division seemed to work well for cross-device interaction when playing cards, as every device served a distinct role that was known to all participants.

Previous studies have examined cross-device interaction in settings where devices were shared (Lucero et al., 2011, Nielsen et al., 2014). Nielsen et al. (2014) explored cross-device interaction using JuxtaPinch, which enabled photo sharing between collocated people. They found that people were generally positive towards sharing personal, private devices *if* sharing with familiar people (friends, family, etc.). Nielsen et al. (2014) also found that people would mostly use and operate their own device when interacting in a large ecology of devices. Synchronization is a related problem when sharing devices between multiple users. As a consequence, this introduces aspects of simultaneous use that can cause problems. Previous studies have pointed to this problem for cross-device interaction, e.g. Lucero et al. (2011), which can be as critical as making the system crash. We did not, however, experience this problem during our second study. We believe that the main reason for this was that the activity of playing cards imposed a natural order of interaction, as players would take turns.

## CONCLUSION

Our work investigated cross-device interaction and in our case we looked at the case of playing card games on mobile devices (mobile phones and tablets). This case works well for studying cross-device interaction as it typically involves more users and a private as well as a shared interaction space (the individual player's hand of cards, and the cards on the table). We identified and compared six different interaction techniques applicable for handheld-to-tablet interaction (*swipe, touch-and-hold,* and *wrist-whip*) or tablet-to-handheld interaction (*directional-swipe, drag,* and *tap-tap*).

Our findings show that mimicking the natural gesture of playing cards in our *wrist-whip* interaction technique was slower than the other techniques and that our participants found it less useful and less natural. We also found that the *swipe* gesture would cause significant problems for our participants in that they often made an interaction error while trying to *swipe* during handheld-to-tablet interaction. Our field-based study showed that in real card game situations, people would sometimes mix up the techniques and thus perform a wrong action in the system. However, our participants appreciated the open nature of the developed card-playing prototype, particularly the regular card playing participants.

## ACKNOWLEDGMENTS

We would like to extend our gratitude to the participants in the two studies.

## REFERENCES


Alsos, O.A. and Svanæs, D. Interaction techniques for using handhelds and PCs together in a clinical setting. In Proc NordiCHI'06. ACM (2006), 125-134.

Boring, S., Jurmu, J. and Butz, A. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In Proc OzCHI'09. ACM (2009), USA, 161-168.

Chapuis, O., Bezerianos, A. and Frantzeskakis, S. Smarties: an input system for wall display development. In Proc CHI'14. ACM (2014), 2763-2772.

Chen, X., Grossman, T., Wigdor, D.J. and Fitzmaurice, G. Duet: exploring joint interactions on a smart phone and a smart watch. In Proc CHI'14. ACM (2014), 159-168.

Chi, P-Y and Li, Y. Weave: Scripting Cross-Device Wearable Interaction. In Proc CHI'15. ACM (2015), 3923-3932

Dix, A., Finlay, J. E., Abowd, G. D., and Beale, R. Human-Computer Interaction (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

Döring, T., Shirazi, A. S., and Schmidt, A. Exploring gesture-based interaction techniques in multi-display environments with mobile phones and a multi-touch table. In Proc AVI'10, ACM (2010), 419–419.

Hamilton, P. and Wigdor, D.J. Conductor: enabling and understanding cross-device interaction. In Proc CHI'14. ACM (2014), 2773-2782.

Houben, S., Frost, M. and Bardram, J.E. HyPR Device: Mobile Support for Hybrid Patient Records. In Proc ITS'14. ACM (2014), 249-258.

Houben, S. and Marquardt, N. WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications. In Proc CHI'15. ACM (2015), 1247-1256

LePage, P. This end up: Using device orientation. http://www.html5rocks.com/en/tutorials/device/orientation/, 2011.

Lowry, R. One-way analysis of variance for correlated samples: k=3. http://faculty.vassar.edu/lowry/corr3.html, 2001.

Lucero, A., Holopainen, J. and Jokela, T. Pass-them-around: collaborative use of mobile phones for photo sharing. In Proc CHI'11. ACM (2011), 1787-1796.

Marquardt, N. Hinckley, K. and Greenberg, S. Cross-device interaction via micro-mobility and f-formations. In Proc UIST'12. ACM (2012), 13-22.

Microsoft. Touch interaction design. http://msdn.microsoft.com/en-us/library/windows/apps/hh465415.aspx, 2012.

Nebeling, M., Mintsi, T., Husmann, M. and Norrie, M. Interactive development of cross-device user interfaces. In Proc CHI'14. ACM (2014), 2793-2802.

Nielsen, H.S., Olsen, M.P., Skov, M.B., and Kjeldskov, J. JuxtaPinch: Exploring Multi-Device Interaction in Collocated Photo Sharing. In Proc MobileHCI'14. ACM (2014), 183-192

Norman, D. A. Natural user interfaces are not natural. interactions 17, 3 (May 2010), 6–10.

Ohta, T., and Tanaka, J. Pinch: An interface that relates applications on multiple touch-screen by pinching gesture. In Advances in Computer Entertainment, A. Nijholt, T. Romo, and D. Reidsma, Eds., vol. 7624 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, 320–335

Rekimoto, J. Multiple-computer user interface: A cooperative environment consisting of multiple digital devices. Proc. CoBuild 1998. Springer (1998), 33-40

Shneiderman, B. Designing the User Interface: International Version: Strategies for Effective Human-Computer Interaction, third ed. Addison Wesley Longman, 1998