CircularSelection: Optimizing List Selection for Smartwatches

Katrin Plaumann **Michael Müller** **Enrico Rukzio**
Institute of Media Informatics
Ulm University, Ulm, Germany
katrin.plaumann@uni-ulm.de michael.mueller-4@uni-ulm.de enrico.rukzio@uni-ulm.de

ABSTRACT
As the availability of smartwatches advances, small round touchscreens are increasingly used. Until recently, round touchscreens were rather uncommon, and so most current user interfaces for small round touchscreens are still based on rectangular interface designs. Adjusting these standard rectangular interfaces to round touchscreens without loosing content comes with loss of precious display space. To overcome this issue for list interfaces, we introduce CircularSelection. CircularSelection is a list selection interface especially designed for small round touchscreens. CircularSelection consists of a ring aligned to the edge of the touchscreen. On that ring, categories of list entries can be preselected. The selected categorie's entries are displayed in the middle of said ring, and can be selected with a tab. In this paper, we explore the design space of CircularSelection by conducting a user study (n = 24) examining its dimensions. We further compare CircularSelection to a traditional smartwatch list selection interface in a second user study (n = 15). Our results clearly show that CircularSelection outperforms traditional smartwatch list interfaces in terms of user preference as well as task completion time (up to 66% faster for large lists and 45% for small lists).

Author Keywords
smartwatch; circular interface; list selection; wearables;

ACM Classification Keywords
H.5.2 Interfaces and Presentation: Input devices and strategies, Interaction styles.

INTRODUCTION
The emergence of smartwatches in recent years poses diverse challenges for user interface and interaction designers. One of the main reasons is their small touchscreen. The resulting interaction space is rather limited, and should be carefully considered, since the fat finger problem and occlusion frequently occur [11]. Interactions are further complicated by varying form factors. In addition to rectangular displays, some Smartwatch vendors imitate the appearance of traditional watches by equipping their smartwatches with round touchscreens. Examples for Smartwatches with round displays are the *Motorola Moto 360*, *LG Watch R* and *LG Watch Urbane*, *Pebble Round Time*, *Samsung Gear S2* and *Samsung Gear S2 classic*.

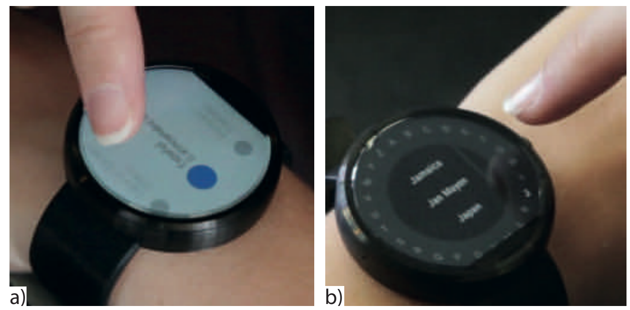

Figure 1. The standard android list interface (a) and CircularSelection (b): CircularSelection is optimized for round displays and uses more of the precious space. This is achieved by using a ring aligned to the edge of the touchscreen for preselecting categories of a list. The list items of the selected category can be selected by taping.

Until the emergence of smartwatches, round user interfaces were rather uncommon for mobile devices. Although vendors equip user interface and interaction designers with styleguides for the new platforms, e.g. the Android design guidelines [1], existing user interfaces were almost exclusively developed with rectangular touchscreens in mind [5, 6, 7, 8, 16]. An example of this is the standard Android list interface depicted in Fig. 1a). Not adjusting rectangular user interfaces leads to wasting expensive space on the yet small display. The largest rectangle fitting into a circle only occupies about 63% of the circle's area. The smartwatch used for evaluating CircularSelection, a *Motorola Moto 360* with a screen diameter of $42\,\text{mm}$, has an available display area of $13.85\,\text{cm}^2$. The largest rectangle fitting into this area occupies only $8.73\,\text{cm}^2$. Therefore around $5\,\text{cm}^2$ of display space are lost if an interface designed for rectangular displays is not adjusted for round displays. The *LG Watch R* has a display diameter of $3.3\,\text{cm}$, resulting in a total display area of $8.55\,\text{cm}^2$ of which only $5.39\,\text{cm}^2$ can be used with a rectangular interface. Since list selection is one key use case for smartwatches [11], and small touchscreens are likely to slow down interaction times for list selection tasks [12], list selection is one of the applications most needing adjustments for small round interfaces.

We therefore propose CircularSelection, a user interface alternative for list selections especially developed for small, round touchscreens. As shown in Fig. 1b), CircularSelection uses the available interaction space by using a ring at the edge of the display. This ring is used to pre-select categories. The list elements of selected

ISWC '16, September 12-16 2016, Heidelberg, Germany

ACM 978-1-4503-4460-9/16/09$15.00
DOI: http://dx.doi.org/10.1145/2971763.2971766

categories are displayed in the middle of the CircularSelection ring. With CircularSelection, we accomplish up to 66% faster selection of list entries. In particular, our contributions are the following:

- An exploration of the design space of CircularSelection.
- An empirical user study exploring several variants of CircularSelection.
- A comparative user study showing that CircularSelection outperforms traditional smartwatch list selection interfaces with regard to task completion time and user satisfaction.

In the following, we will first discuss related work before we explain CircularSelection in detail. Subsequently, we will present the user studies and the limitations of CircularSelection.

RELATED WORK

Designing interfaces for round smartwatches

Interaction with smartwatches is a very active research field within Human Computer Interaction. Smartwatches are topic of interest for varying contexts, and several applications for smartwatches have been developed and evaluated in the past. Among these projects are applications augmenting interactions in offices [5], handling of notifications [6], textentry [7], combined use of smartwatch and other mobile devices [8], and navigation [16]. However, all of these applications have been developed with rectangular shapes of interfaces in mind, and depending on the application could be hard to migrate to round smartwatches.

Lyons [9] provides an analysis of usage of digital watches which also serves as information basis for further smartwatch designs. Due to the variety of styles and forms of self-expression linked with watches, they suggest that smartwatches should support individual styles and designs. Exploring the design space of smartwatches, Xu and Lyons developed two smartwatch prototypes with round faces [17]. Albeit their prototypes have round faces, and the designs are adapted to that form factor, Xu and Lyons have a strong focus on displaying notifications, while we focus on list selection. Ashbrook et al. investigated how the trade-off between button area and display area affects the error rate of selections on small, round touchscreens [4].They derived a mathematical model for error rates of buttons aligned to the rim of a round watch. While Ashbrook at al. focus more on basic principles, we address a specific use case, list selection.

List selection for smartwatches

Perrault et al. used a position sensor build in a watch's strap to navigate lists [11]. While this approach enlarges the interaction space by leveraging other components than the touchscreen, we focus on using the available interaction space and thus being more independent of specific hardware. With their cylindrical smartwatch prototype, Strohmeier et al. analysed the effects of different display sizes on list selection tasks [12]. Their prototype consisted of a flexible touchscreen wrapped around user's wrist. Different display sizes were simulated by displaying rectangular user interfaces of different dimensions. The whole area of the cylindrical smartwatch served as input space for all display sizes, while we focus on using the display space as input space.

Current native Android list selection interfaces show three items placed amongst each other [2]. List interfaces of other Smartwatch operating systems like Pebble and Tizen follow the same patter [10, 14]. CircularSelection avoids this concept's limitations by enabling preselections of categories. To the best of our knowledge, the user interface design elements closes to our concept are the *More Option* of the Pebble smartwatch SDK [13] and the *Friend Carousel* of Apple's Watch [3]. With the first, several menu items represented as icons are arranged around the edge of the touchscreen. Those icons can be selected with tapping or by a moveable bezel. With the latter, contact information of friends are arranged in a circle, allowing users to organize their contacts. Contacts can be selected using the watch's crown and tapping. CircularSelection differs from both by focusing on optimizing list selection and on exploring the design space.

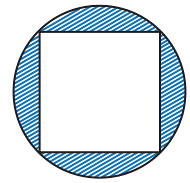

Figure 2. Schema of a rectangular user interface fitting into a round display. The blue hatched areas are those not used by the user interface.

INTERFACES FOR ROUND SMARTWATCHES

Round interfaces are relatively new for mobile devices. Before the emergence of smartwatches, almost all touchbased interfaces were rectangular. Furthermore, displays used in non mobile scenarios are mostly rectangular. Considering this, it is not surprising that the majority of user interfaces is designed for rectangular interfaces. Besides the overall layout of interfaces, the interface elements themselves are often rectangular or contain straight lines. Examples include left or right justified text blocks, text fields, buttons, drop down menus, tables, menu bars, cards and pages.

Porting interfaces based on rectangular designs to round smartwatches poses certain challenges, mainly due to the largest rectangle fitting into a circle only occupying 63% of the circle's area. Therefore, using the same user interface for both round and rectangular smartwatches comes with loss of precious interface space. Figure 2 depicts a circle with a square fitting into it. The circle represents a round smartwatch touchscreen, while the square represents a rectangular user interface. As can be seen, the blue hatched area of the circle is not used by the user interface. Usually, this space is filled with background rather than interface elements. This poses the problem that precious space on the yet so small display is not used. Also, not adequately incorporating this space into the user interface design might not be as visually appealing as a user interface designed for round displays.

There are several possible ways of overcoming this issue and designing user interfaces for round, small displays. For example, text could be centred vertically, so that it floats to either sides. Straight lines could be replaced by curved ones, and edges could be avoided. Important interaction elemenets cold rather be placed at the left and right side as well as above and below the centre instead of in corners of a (thought) rectangle. Other solutions are rounded buttons at the bottom, left and right side, progress indicators aligned to the edge of the display, and aligning menu items to the edge of the display [14]. With CircularSelection, we propose a list selection user interface especially designed for

Figure 3. The two possible segement sizes *fixed* (a) and *flexible* (b). While with *fixed segment size*, all segments have the same size, *flexible segment size* increases the space used by segments with categories with more entries. As can be seen in image (b), the segments of the letters *A*, *B*, and *S* occupy more space than the segments of the letters *U*, *V*, and *W*. This is due to more list entries starting with either *A*, *B*, or *S* than *U*, *V*, or *W*.

round smartwatches, contributing to the user interface elements for round small touchscreens.

CIRCULARSELECTION

The basic concept of CircularSelection is to use a ring aligned to the bezel (as depicted in Fig. 1b) to preselect a category of the list, e.g. an initial character. The ring contains segments for each category (e.g. the initial characters) of the list entries. There is always one active segment. Within the inner area of the ring all items from the list of the selected (in other words, active) category are displayed. The inner area resembles a traditional scrollable list view, where items can be selected with a single tab. The CircularSelection ring occupies 40% of the display, the inner area 60%. Albeit being designed for round interfaces, preselection of categories could also optimize list selection on rectangular interfaces. Therefore used segments could also be aligned to the rim of the display. Instead of showing the inital letters of categories in the segments, what strongly implies usage of alphabetical lists, icons could be used. This way, CircularSelection could not only applied to alphabetical lists like contactbook entries, by also for settings menus or app lists.

During the design process, several possible alternations to this basic concept were identified, namely the segment size, selection method, and direction of alphabet. In the following, each is explained in detail and it is outlined why those alternations were thought to be beneficial.

Segment Size

Two alternative segment sizes were considered: *fixed* and *flexible size*. While segments with *fixed size* are all of equal size, the size of segments with *flexible size* varies according to the number of entries in the segment's category. This way categories with more entries are represented by larger segments. Figure 3 shows both variants. As can be seen in 3b, the segments of the letters *A*, *B*, and *S* occupy more space than the segments of the letters *U*, *V*, and *W*. This is due to more list entries starting with either *A*, *B*, or *S* than *U*, *V*, or *W*. To ensure that each segment is selectable, a minimum size of 7° for each segment was used. The remaining degrees (182°) were distributed according to the number of entries in each category.

The idea behind the segment size increasing with an increased number of entries in the segments' category was to fasten interaction. This consideration resulted from larger targets being generally easier to hit, and segments with more entries being hit more often. Assuming that categories with more entries are also

Figure 4. The three selection methods *fixed ring* (a), *moveable ring* (b), and *jumb back ring*. As can be seen, with a fixed ring the preselection is done with a tab on a segement of the ring. The actual item can be selected by scrolling and tapping in the list in the middle of the ring. The moveable ring must be moved so that the segment of the wanted category is in the 12 o'clock position. The same is true for the jump back ring, but here the ring jumps back into its original position.

more frequently selected, this would overall increase selection time by reducing false preselections and resulting corrections. One limitation of *flexible segement size* is that it would be harder to use icons, since they would have to be adjusted to each segement size.

Selection Method

For selecting a category (e.g. an initial character), the three methods *fixed ring*, *movable ring* and *jump back ring* were identified. With a *fixed ring*, depicted in Figure 4a), a category can be selected by tapping the corresponding segment. Instead of tapping, a moveable bezel or a hardware crown could be used to chose a segment. The position of the segments are not changeable, in contrast to the *movable ring* shown in Figure 4b). Here, the ring has to be moved until the segment of the wished category is in the 12 o'clock position. This could be achieved by using a thouchscreen, a moveable bezel were the CircularSelection ring as aligned with the bezel's position, or by turning a crown. The 12 o'clock position is visually highlighted. Upon a segment entering that position, the list in the inner area is updated. We included a moveable ring because it allows users to let their finger be placed on the touchscreen while se-

Figure 5. Both directions of alphabet: *clockwise (or forward)* (a) and *counter-clockwise (or backward)* (b).

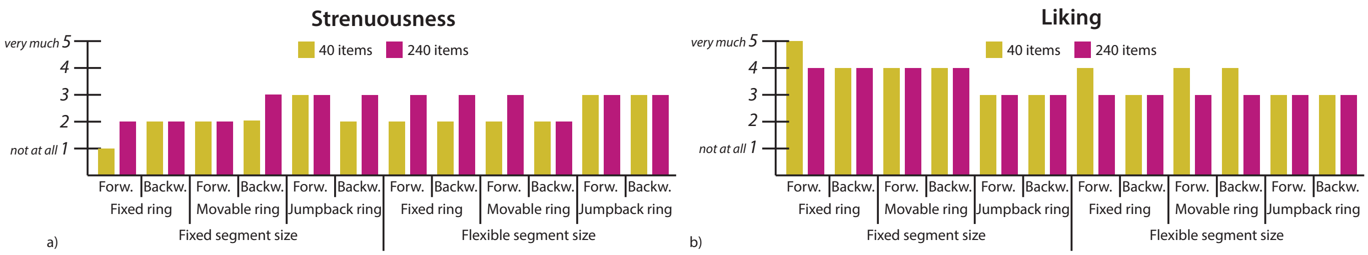

Figure 6. Participants' median ratings regarding the strenuousness of each condition (a) and how much they liked the condition (b). Ratings were given on five point likert scales, with 1 being "not at all" and 5 being "very much". Each bar represents one condition. Each condition is defined by their alphabet direction (forward or backward), selection method (fixed ring, moveable ring, and jump back ring), and segment size (fixed or flexible) as well as the length of the list it (40 items or 240 items).

lecting a category. This way, the finger is stabilized and guided by the display, potentially making interaction faster and easier as well as avoiding corrections. The *jump back ring* (cf. 4c) also requires to move the wished segment in the 12 o'clock position, however, instead of staying in that position, the displayed ring "jumps back" into its original position. This ensures that each segment always has the same starting position. In contrast to *moveable ring*, users always know were a segment is placed on the watch. This is supposed to decrease time needed to search for a segment, and thus fasten interaction. Figure 4c) shows the interaction flow.

Direction of Alphabet
For alphabetical lists, there are two possibile alphabet directions to arrange the segments on the ring: *clockwise* (forward, cf. 5a) or *counter-clockwise* (backward, cf. 5b). Counter-clockwise alphabet was included because preliminary tests revealed that especially when using the moveable ring, users tend to prefer the alphabet displayed in counter-clockwise direction.

USER STUDY 1 - EVALUATING DESIGN ALTERNATIVES
The first study served to evaluate the above described alternations to the basic concept of CircularSelection. The results show which combinations of segment size, selection method and direction of alphabet allow fast and accurate list selection.

Participants
24 participants (8 female) took part in the study. The average age was 26.5 (SD= 2.9). Three participants were left-handed. All reported experience with touch screens and had an ongoing or completed academic education. All participants received €8 compensation.

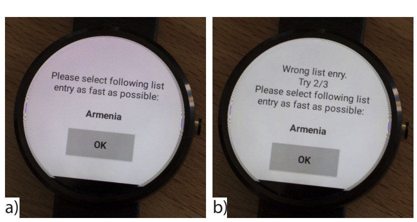

Figure 7. The used dialogs to indicate the list item participants should select (a) and, in case of a falsely selected item, repeat the item to select and the number of tries left (b).

Study design
As we aimed to evaluate the design alternatives resulting from the above mentioned possible combinations, we treated segment size, selection method and direction of alphabet as independent variables. Furthermore, two different list sizes (small and large) were used, resulting in a $2 \ x \ 3 \ x \ 2 \ x \ 2$ factorial within subject design. Counterbalancing was applied to avoid carry over effects. The small list contained 40 items (names of states), the large list contained 240 items (countries and their dependencies). The list sizes were conceived based on the ones used by Perrault et al. [11]. However, we referred from using three list sizes because pilottests revealed that this would unduly increase the duration of a test session. Instead, we adjusted the size of the small list to 40 entries. Participants' task was to select items from the displayed lists. As did Perrault et al. [11], we choose several distances between items. Those distances were used to chose the items participants should select. The distances were chosen randomly and added to the item currently selected (or the first item, for the first selection task). Distances were 5, 10, 20, and 39 for the 40 item list, and 5, 10, 20, 40, 80, and 159 for the 240 item list.

Hypotheses were that (1) a movable ring that follows the finger yields faster selection results than a fixed ring, due to movements being guided and stabilized by the touchscreen. (2) The dynamic chamber sizes will yield faster completion time, due to increasing accuracy for preselection of categories with more entries.

Apparatus
All design alternatives were implemented using the Android Wear platform and displayed on a Motorola Moto 360 smartwatch. The used watch has a display diameter of 4 cm and a display resolution of $320 \ x \ 240$ pixel. Note that the display is not completely usable as interactive area, since there is a small black bar on the bottom of the touch screen. This small black bar contains the touch and light sensors and does not recognize touch gestures.

Procedure
After giving informed consent and completing an initial survey, the participants were handed the smartwatch. They were asked to not use any table or surface to support their arm. To further prevent the placement of arms on furnitures participants were placed on a chair without arms. It was tolerated if they used their body as support. This position was chosen to simulate a mobile usage scenario, where users might sit, but would not necessarily have other means to support their arms than their own body.

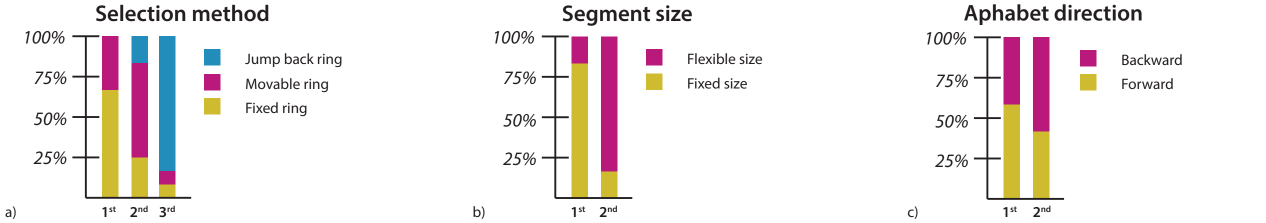

Figure 8. Participants rating for selection method (a), segment size (b), and alphabet direction (c). Participants were asked to rank the variants regarding their preferred use. The bars resemble the rank of the characteristics (1st, 2nd, and 3rd best), the colours the characteristics and the proportion of the colour of the bar the percentage of users rating the characteristic this way.

Participants were advised to wear the smartwatch on the arm of their non-dominant hand.

After participants received instructions and signalled that they were ready, the 24 conditions were tested in predefined order. For each alternative, participants had to pass a training phase. The training phase consisted of successfully selecting three items. Afterwards, participants had to select five items, which were used for the analysis. The item to select was indicated on a special screen on the watch (cf. 7a), and had to be confirmed by tapping a button. Task completion time was measured from this confirmation until the successful selection. For every selection task, participants had three tries. After an unsuccessful try, the remaining tries as well as the item to select were shown on the watch and had to be confirmed again (cf. 7b).

After each of the 24 conditions participants were asked to rate how strenuous they found the condition and how they liked the condition on five point likert skales ranging from “1 - not at all” to “5 - very much”. After completing all 24 conditions participants were asked to rank the levels of each independent variable, ranging form the level they most preferred to the level they least preferred.

Results

Participants' rating

Participants' median rating regarding strenuousness and liking for each condition are depicted in Fig. 6a and 6b respectively. Ratings were given on five point Likert scales, with 1 being “not at all” and 5 being “very much”. Each bar represents one condition. As can be seen, conditions with fixed segment sizes are rated rather less strenuous (61.54% of participants rated conditions involving fixed segment sizes strenuousness with 2 or less) and more liked (72.72% rated such conditions' liking with 4 or more) than dynamic segment sizes. Conditions incorporating the fixed ring were rated less strenuous (46.15% of conditions rated 2 or lower) and more liked (45.45% of conditions rated 4 or higher) than conditions involving moveable ring (38.46% and 54.54% respectively) and jump back ring. For alphabet direction, ratings are more evenly distributed, though clockwise alphabet direction is both most liked (54.54% of conditions rated 4 or higher) and rated least strenuous (46.15% of conditions rated 2 or lower). Skilling Mack tests revealed that the overall differences between conditions were significant for strenuousness ($\chi^2(23) = 138.17$, $p < 0.001$) and for liking ($\chi^2(23) = 132.14$, $p < 0.001$).

Participants rating regarding their favoured form of selection method, segment size and alphabet direction is depicted in Fig. 8. The bars resemble the rank of the characteristics (1st, 2nd, and 3rd best), the colours the characteristics and the proportion of the colour of the bar the percentage of users rating the characteristic this way. As can be seen, a fixed ring was preferred by most participants followed by the moveable ring, while the jump back ring was rather disliked. Also, the size of the sections was preferred to be fixed. Regarding the alphabet direction, ratings are again not so clear. Here, only 58% preferred forward direction over backward direction.

Error rates

All participants eventually selected all list entries correctly within the given number of tries. Overall error rates were rather low. Figure 9 shows the average number of tries needed to select the correct item. As can be seen, items could almost always be se-

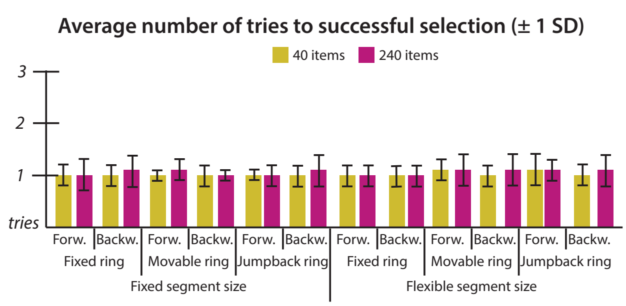

Figure 9. The average number of tries needed to successfully select the given list entry for all conditions. Error bars show two standard deviations). Each bar represents one condition. Each condition is defined by their alphabet direction (forward or backward), selection method (fixed ring, moveable ring, and jump back ring), and segment size (fixed or flexible) as well as the length of the list it (40 items or 240 items).

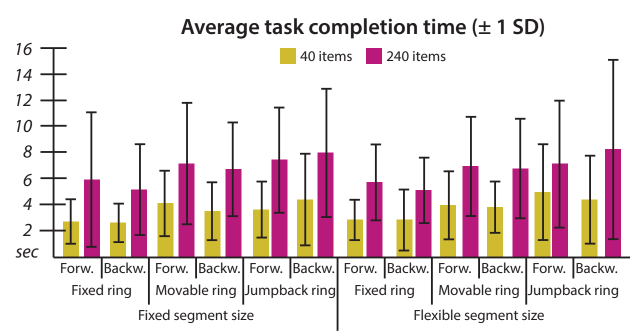

Figure 10. The average time it took participants to successfully select an item (error bars show two standard deviations). Each bar represents one condition. Each condition is defined by their alphabet direction (forward or backward), selection method (fixed ring, moveable ring, and jump back ring), and segment size (fixed or flexible) as well as the length of the list it (40 items or 240 items).

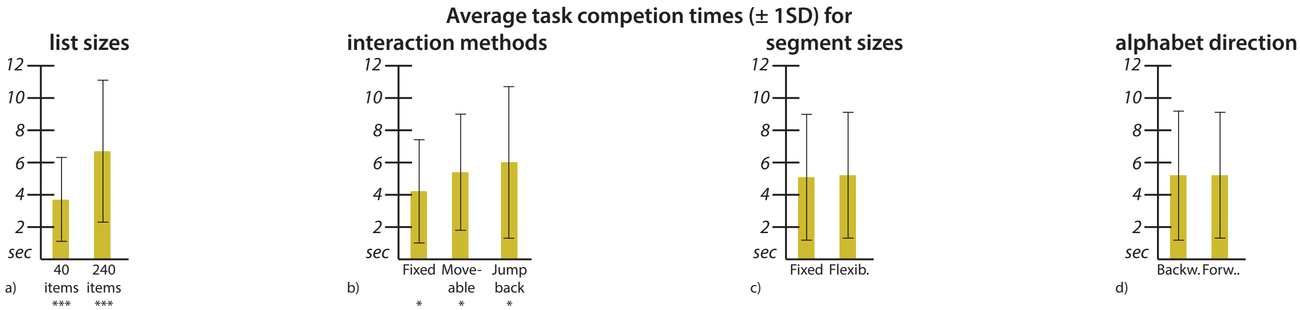

Figure 11. The average time it took participants to successfully select an item split for the four used factors list size (a), selection method (b), segment size (c), and alphabet direction (d). Error bars show two standard deviations. Significances are indicated as follows: * for p-values lesser than 0.05 and * for p-values lesser than 0.001.**

lected with the first try. The only significant main effect a repeated measures ANOVA could reveal was for list size ($F_{1,2377} = 4.66$, $p < 0.05$). For the large list, little more tries were needed than for small list (1.06 tries (0.26 SD) and 1.04 tries (0.19 SD)).

Task completion time

Fig. 11 shows the average task completion times for list size, selection method, segment size and alphabet direction. As can be seen, shorter lists result in shorter interaction times. Fixed ring outperforms moveable ring, which still yields faster selection times than jump back ring. Also, fixed sized segments allow faster selection times. Backward displayed alphabets are slightly faster than forward displayed alphabets. The average task completion times are depicted in Fig. 10. Each bar represents one condition. A repeated measures ANOVA revealed significant main effects for list size ($F_{1,2377} = 436.35$, $p < 0.001$) and selection method ($F_{2,2377} = 58.98$, $p < 0.001$). Post hoc comparisons (t-tests, Bonferroni correction used) for selection method revealed significant differences between all selection methods ($p < 0.05$). Interactions were found to be significant for selection method and alphabet direction ($F_{2,2377} = 3.48$, $p < 0.001$). When selecting categories with the jump back ring, participants were faster with the alphabet displayed in forward (e.g., clockwise) direction (6279.94 ms (5188.413 SD) for backward alphabet and 5819.26 ms (4094.66 SD) for forward alphabet), while when using fixed ring and moveable ring, participants were faster with the alphabet being displayed in backward direction (3968.96 ms (2800.53 SD) for backward and 4332.34 ms (3501.07 SD) for forward and 5233.95 ms (3361.67 SD) for backward and 5537.65 ms (3789.27 SD) for forward, respectively).

Discussion

Our results clearly show that fixed ring outperforms moveable and jump back ring regarding task completion time and user preferences, although the fixed-ring concept has problems with occlusion and the fat finger problem, similar to the problems described by Vogel et al. [15]. However, we found this to be less of a problem since the inner area always indicates which initial character had been chosen. Since the inner list was updated as soon as a segment was touched, but the actual selection only occured when the finger was lifted, users could keep their finger on the screen as long as needed to find the intended category. Keeping the finger on the screen could have provided more stability and thus prevented errors. Regarding the moveable ring and the jump back ring, both users ratings and task completion time show that the moveable ring performs better than the jump back ring. Regarding the segment size, the ANOVA did not reveal a significant effect, however users ratings are rather clear, and speak for fixed segment sizes. One the one hand, dynamic segment sizes may fasten overall interaction time due to segments representing longer lists are larger and thus easier to tab. On the other hand, however, varying segment sizes might look disordered and thus confusing users. Since we have not found a significant effect, we should rely on users ratings classifying fixed segment sizes as better than dynamic segment sizes. Less clear are the results for alphabet direction: neither task completion time nor users ratings show a clear tendency. Albeit being rated a little better by users, forward alphabet direction is not significantly faster than backward alphabet direction. On the contrary: for fixed ring and moveable ring backward alphabet direction was faster. We thus decided to include both alphabet directions in the second user study, comparing fixed ring interaction with fixed segment sizes and both alphabet directions with a traditional smartwatch list interface. All tested alternatives had relatively small error rates, whith the average number of tries needed to successfully select an item being only marginally above one.

USER STUDY 2 - COMPARISON WITH TRADITIONAL LIST INTERFACE

In this study the two best design alternatives retrieved from the previous study were compared with the traditional Android Wear list interface.

Participants

15 participants (6 female) took part in this study. The average age was 27 (SD= 5.16). All but one participant were right-handed, though eventually three participants decided to wear the watch on the right hand due to being used to wear their watch on that hand. 12 participants wore the smartwatch on the left hand. All reported experience with touch screens, but only three participants reported some experience (1 year) with a smartwatch. All participants had an ongoing or completed academic education. All participants were rewarded with €5 .

Apparatus

The same apparatus as in study one was used.

Study design

As we aimed to compare the two best forms of CircularSelection retrieved from the first study with the traditional Android Wear

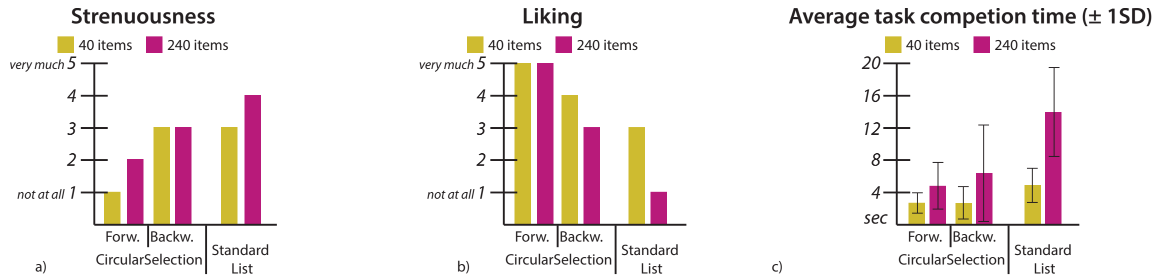

Figure 12. Participants ratings for streneousness (a) and liking (b), as well as task completion time in seconds (c) for CircularSelection with forward and backward alphabet direction as well as for the baseline technique (standard Android list interface as shown in Figure 1a). Error bars represent two standard deviations. Each bar represents one condition.

list user interface (cf. 1a), our first independent variable was the used list selection interface. Second independent variable was list size, resulting in a 3 x 2 factorial within subject design. Counterbalancing was applied to avoid carry over effects. List sizes and choice of itmes were the same as in study one.

Procedure

The same procedure as in the first study was applied, with the number of conditions tested being adjusted to 6 based on our study design.

Results

Participants' rating

After testing each condition, participants were asked to rate the strenuousness and their liking of the condition on five point Liker scales, with 1 being "not at all" and 5 being "very much". Fig. 12a and Fig. 12b show the results for strenuousness and liking, respectively. As can be seen, both CircularSelection interfaces are more liked and rated less strenuous, with the variant with forward alphabet direction being more in favour. Asked about their preferred selection interface after all conditions, all participants chose CircularSelection with forward displayed alphabet, followed by CircularSelection with backward displayed alphabet, and the traditional list selection interface.

Error rates

As in the first study, error rates were rather low and almost all list entries were successfully selected with the first try. Figure 13 depicts the average number of tries for each condition. A repeated measures ANOVA did not reveal significant effects.

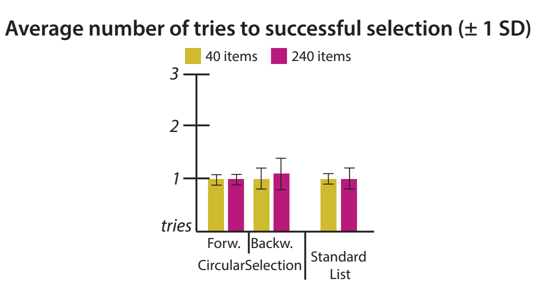

Figure 13. The average number of tries needed to successfully select the given list entry for all conditions. Error bars show two standard deviations. Conditions were CircularSelection with alphabet displayed in forward (clockwise) and backward (counter clockwise) direction and the standard Android list user interface.

Task completion time

Task completion times for all conditions are depicted in Fig. 12c. As can be seen, both CircularSelection interfaces outperform the traditional smartwatch list selection interface, while CircularSelection with forward alphabet direction is yet faster than with backward alphabet direction. In total, the traditional list selection interface achieved a task completion of 5100.19 ms (2236.23 SD) for small and 14664.45 ms (14664.45 SD) for large lists, CircularSelection with backward alphabet direction 2790.11 ms (2103.29 SD) and 6662.65 ms (6289.53 SD) respectively, CircularSelection with forward alphabet direction 2802.05 ms (1322.12 SD) and 5047.31 ms (3042.30 SD) respectively. A repeated measures ANOVA revealed significant main effects for list size ($F_{1,445} = 207.10$, $p < 0.001$) and interaction method ($F_{2,445} = 96.34$, $p < 0.001$). List size and interaction method have a significant interaction ($F_{2,445} = 43.86$, $p < 0.001$). Post hoc tests for interaction method (t-tests, Bonferroni correction used) revealed significant differences between the traditional list selection interface and both CircularSelection interfaces ($p < 0.001$).

Discussion

The results show that both CircularSelection interfaces clearly ourperform traditional smartwatches. The best CircularSelection variant, using a fixed ring with fixed segment sizes and forward alphabet direction, allows 66% faster selection for large lists and 45% faster selection for small lists. Even though the flat tire (the black bar at the bottom of the screen, cf. 4) of the Moto 360 interferes with our ring, we still found our ring concept being superior to a native list view. Additionally, user preferred CircularSelection over the Android list selection user interface, finding it more likeable and less strenuous.We showed that optimizing list interfaces for round form factors not only fastens interaction, but also increases user satisfaction. Both are important factors for successful interactions with smartwatches.

It remains unclear why in the first study some participants rated the fixed ring with a backward alphabet higher than a forward alphabet. A possible explanation could be the larger sample size (n=15 vs n=24), though we rather suspect that some participants were confused by rating all 12 versions against each other.

LIMITATIONS

There are some limitations to our concept, which we want to address here. Our results show that CircularSelection outperforms the standard Android list selection interface more strongly with large lists (66%) than with small lists (45%). This could be explained by a tradeoff between the time needed for the preselection and for scrolling in the list. For short lists, scrolling could be faster than preselection. Considering this and that CircularSelection uses preselection, traditional list selection user interfaces could perform better on very small lists, were users exactly know were certain list entries are located. The study was conducted while participants were sitting, and it is still left to explore if the results are replicable for other conditions like standing or walking. Also, our study was conductuted in a rather controlled environment, how CircularSelection performs in real world settings is still left to explore.

The Moto 360, which we used for our prototype, contains a black bar (the "flat tire", displayed in Figure 4). This is the place where the light sensor is located and it overlapped our ring. The comments from participants showed that they found this black bar irritating. But even though it affected the fixed ring most (since the movable ring could be dragged so that the characters behind the black spot got visible), the fixed ring was still the most preferred interaction technique, and though not being displayed as a perfect ring, CircularSelection outperformed the traditional Android list selection interface.

Incorporating hardware components, like a tactile crown as used on the Apple Smartwatch or a moveable bezel as used on the Samsung Gear S2, may yield other results for both traditional list selection interfaces and CircularSelection.

CONCLUSION

The emergence of Smartwatches poses new challenges for designing user interfaces for small touch interfaces. Adding to the complexity of designing for ultra small displays, designing for round interfaces gains more and more importance since smartwatch vendors started to use round touchscreens. We introduced CircularSelection, a list selection concept suited for ultra-small round touch interfaces with a round form factor.

We explored the design space by implementing a prototype and conducting a study to examine which features from the design space work best in terms of user preference and performance. Our study showed that a fixed ring with discrete touch gestures (selection by tapping) is a clear winner in terms of task completion time, user preference and strenuousness.

Furthermore, we conducted a second user study to compare this best version of CircularSelection with the standard Android list user interface. This study showed that our CircularSelection concept is superior in terms of user preference and strenuousness as well as task completion time.

ACKNOWLEDGMENTS

This work was conducted within the Emmy Noether research group "Mobile Interaction with Pervasive User Interface" and SFB/TRR 62 DFG.

REFERENCES

1. Android Developers: Android Wear. `http://developer.android.com/design/wear/index.html`. Accessed: 2016-02-10.
2. Android Developers: Creating Lists. `http://developer.android.com/training/wearables/ui/lists.html`. Accessed: 2016-03-22.
3. Apple Inc. Get in touch with friends - Apple Watch User Guide. `http://help.apple.com/watch/?lang=en#/apd920c18146`. Accessed: 2016-07-08.
4. Ashbrook, D., Lyons, K., and Starner, T. An investigation into round touchscreen wristwatch interaction. In *Proc. of MobileHCI*, ACM (2008), 311–314.
5. Bernaerts, Y., Druwé, M., Steensels, S., Vermeulen, J., and Schöning, J. The office smartwatch: Development and design of a smartwatch app to digitally augment interactions in an office environment. In *Proc. of DIS Companion*, ACM (2014), 41–44.
6. Bolle, L., De Croon, R., and Duval, E. Optimizing smartphone micro-usage with smartwatch notifications. In *Proc. of MobileHCI*, ACM (2015), 947–950.
7. Chen, X. A., Grossman, T., and Fitzmaurice, G. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proc. of UIST*, ACM (2014), 615–620.
8. Chen, X. A., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: Exploring joint interactions on a smart phone and a smart watch. In *Proc. of CHI*, ACM (2014), 159–168.
9. Lyons, K. What can a dumb watch teach a smartwatch?: Informing the design of smartwatches. In *Proc. of ISWC*, ACM (2015), 3–10.
10. Pebble Developers. Recommended guidelines and patterns. `https://developer.pebble.com/guides/design-and-interaction/recommended/#list-options-with-a-menu`. Accessed: 2016-04-21.
11. Perrault, S. T., Lecolinet, E., Eagan, J., and Guiard, Y. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In *Proc. of CHI*, ACM (2013), 1451–1460.
12. Strohmeier, P., Burstyn, J., and Vertegaal, R. Effects of display sizes on a scrolling task using a cylindrical smartwatch. In *Proc. of MobileHCI Adjunct*, ACM (2015), 846–853.
13. Tizen Developers. More option. `https://developer.tizen.org/development/ui-practices/native-application/efl/ui-components/wearable-ui-components/more-option`. Accessed: 2016-04-21.
14. Tizen Developers. UI components for wearable app design. `https://developer.tizen.org/design/wearable/ui-components`. Accessed: 2016-04-21.
15. Vogel, D., and Baudisch, P. Shift: a technique for operating pen-based interfaces using touch. In *Proc. of CHI*, ACM (2007), 657–666.
16. Wenig, D., Schöning, J., Hecht, B., and Malaka, R. Stripemaps: Improving map-based pedestrian navigation for smartwatches. In *Proc. of MobileHCI*, ACM (2015), 52–62.
17. Xu, C., and Lyons, K. Shimmering smartwatches: Exploring the smartwatch design space. In *Proc. of TEI*, ACM (2015), 69–76.