Int J Software Informatics, Volume 9, Issue 1 (2015), pp. 23–36
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

Ember: A Smartphone Web Browser Interface for the Blind

Ruchika S, Susmitha Pulakhandam, Isha Singh Jassi, and G Ram Mohan Reddy

(Department of Information Technology, National Institute of Technology Karnataka Surathkal, Mangalore, India)

Abstract Ember is a smartphone web browser interface for the blind. It was designed exclusively for the blind, keeping in mind their needs and requirements. With large screen targets and easy Braille based text entry, Ember aims to make browsing an enjoyable experience for the blind user. Furthermore, Ember has a verbal command option for easier input. A simple way to find targets on the interface has been incorporated with Ember, by using a screen guard with projections at six points that correspond to the six dots used in Braille. After testing, it was found that twenty-five out of twenty-eight volunteers, on an average, could navigate faster using the Ember audio navigation method than the traditional web browser, and the learning curve for the Ember interface is less steeper than the learning curve for the traditional browser. The Ember keypad was also tested against the QWERTY keypad and the SmartBraille, a well-recognized keypad for Braille-based text entry, and was found to be more intuitive and natural to use.
Key words: HCI; smartphone interface; Braille; audio hepatic feedback; multi-touch screen

Ruchika S, Pulakhandam S, Jassi IS, Reddy GRM. Ember: a smartphone web browser interface for the blind. *Int J Software Informatics*, **Vol.9, No.1 (2015): 23–36. http://www.ijsi.org/1673-7288/9/i206.htm**

1 Introduction

In today's technologically advanced world, touch interfaces have become significant and are a necessary part while designing any new interface for users. We have ATM machines, personal computers, smartphones, etc. all using this vital interface. Smartphones, in fact, have become an indispensable part of our lives. A smartphone not only has the features of the traditional mobile phone, but also those of other devices such as a media player, a camera, GPS, etc. Functionalities of many devices are available within a single device and hence it becomes important that it is available to all types of users. The blind are at a loss here and hence it is required that novel approaches be made to provide these functionalities to these users.

The touch interface can be evolved in a way such that the blind can use it effectively. The traditional touch and locate paradigm that they follow cannot be applied with touch interfaces of a smartphone. To improve the usability quotient for a blind user haptic and audio feedback can be provided for every touch on the

Corresponding author: G Ram Mohan Reddy, Email: profgrmreddy@gmail.com
Received 2014-29-12; Revised 2015-04-30; Accepted 2015-04-30.

interface[11]. The interface design had to be intuitive, quick to learn and easy to use, this required analysis and insight.

For using internet on their desktops and personal computers the blind use screen reading software such as JAWS and the like. A disadvantage with the screen readers being that the user becomes dependent on how the website is structured as the screen readers read out every line on the website line by line in a linear fashion. Also, screen readers are not portable and meant only for the desktop personal computers as of now. Another approach is using Braille readers, these readers convert every line of text on a website into Braille for the user to read line by line again. This provides a familiar setup to the blind user, but may not be advisable when the user doesn't want to read line by line. The choice to take selected information from a webpage does not lie with the user. Table 1 shows the problems associated with each approach. The Ember interface hopes to eradicate these issues by providing tactile method of interaction which comes most naturally to the blind.

Table 1 Comparison of the strengths and weaknesses of Braille readers and screen reading software

Approach	Strength	Weakness
Braille Readers	Incorporates Braille and tactile; familiar	Can only read one line at a time, bulky, no information about layout of the page
Screen Reading Software	Easy to use, knowledge of Braille is not a prerequisite	Effectiveness depends on the website developer can be slow and inflexible

It was understood that the blind could use the edges of the smartphone as bearings to locate targets on the screen and hence the targets for the interface were designed in such a way that they were easily locatable. The important menus were all located on the four corners of the smartphone screen such as the tools, backward, forward, home, audio input menu, etc. This arrangement left a huge middle space for the text input, the user could touch anywhere within this area and the Ember keypad would pop up for receiving the Braille input. Unlike traditional QWERTY keypad the keys and targets in the Ember interface were large and easier to find and use. The Ember keypad also provided audio input at all stages.

The key contribution of our work is the ability this interface provides to blind users with respect to navigating around the web on their touch screen mobile phones. Web developers and website designers are usually so engrossed in designing a website that is compatible with many different browsers and different platforms that they keep designing websites compatible with screen guards or Braille readers and this is where Ember comes in. Ember would help to the Blind users to use a web browser more easily and input text more easily. Ember would help users navigate websites better on their smartphone.

The rest of this paper has been organized in the following manner. Section 2 holds related work that has been previously carried out in this regard. It also compares and contrasts our proposed work and the various literary works mentioned in this section. Section 3 describes in great detail the methodology followed and the working of the web browser interface. Section 4 discusses the strengths & shortfalls of implementation details with tests carried out and the corresponding results & analysis.

It also holds the tests and results with the analysis. Section 5 deals with concluding remarks and Section 6 highlights the future directions of the proposed work to be extended to overcome the various shortcomings that were observed in the proposed work.

2 Related Work

Ember extends previous research on the accessibility of touchscreen interfaces by providing a more suitable input method and navigation techniques which address the concerns expressed by blind users.

2.1 Implications for design

The *Slide Rule*[1] research proposed ideologies that helped shape the design of Ember's interface. Blind users often carry a multitude of devices even though most devices overlap in function because a particular device may provide a suitable interface for certain functionalities, but prove poor in another respect. The research showed that the difficulties blind users have include difficulty in learning where objects were located on the screen. Some informants also mentioned that they were afraid of making mistakes that could not be undone.

The *Slide Rule* incorporates various gestures, the L-select gesture in particular, to navigate through lists, etc. Based on this feedback Ember was designed for the touchscreen mobile because the touchscreen can incorporate the functions of several devices into a single device. The interface should allow the users to interact with familiar layouts. Ember's skeleton layout stays the same throughout all navigation phases so the learning process is made easier. The *Slide Rule* research also showed that touchscreen interfaces must be easy to explore and minimize the need to find a particular point on the screen through trial and error. Ember's layout uses the edges of the keyboard as bearings, provides large buttons and a very large area to prompt text input. As the user passes over a button or text field with his finger, the label of the widget is read out to him. The need to pinpoint a specific area of the screen is almost negligent when the verbal command option is used. The consistent audio feedback and verbal command options make Ember's design functionally more apt then the *Slide Rule*.

2.2 Acknowledging the need for a different method of touch based text entry

In Ref. [2], four different methods of text input were analysed. QWERTY[3, 9], MultiTap[6], NavTouch[4] and BrailleType[5] were compared by testing their effectiveness across users of different genders, age, spatial ability, verbal IQ and Braille reading and writing in words per minute. The study aims to assess the advantages and limitations of different touch-based text-entry approaches, and to show that different blind users with different individual attributes can benefit from one method over another. The results showed that there was not a consensus on most methods except that BrailleType users collectively agreed that they would use the system.

BrailleType takes advantage of the capabilities of those who know the Braille alphabet. The touchscreen is divided into six large cells, each representing the dot positions. Users can explore painlessly as audio feedback is provided and double tap is required to select a target. By reducing the number of on screen targets, as compared to the QWERTY pad, for example, less stress is placed on the blind user. Although BrailleType was the method which posed the least number of errors, there were still errors. The errors were mostly caused by the timeout mechanism of the keypad. The keypad did not allow experienced users to type faster and so the wrong letter was registered when the user tried to move on to the next letter. It made the process of text input very slow.

The Ember Keypad does not have this problem there is no need for a timeout. The keypad is designed in the way that all the targets to be pressed at once and are registered simultaneously using multi-touch. The phone is held horizontally with the touchscreen facing away from the user such that they can use two hands simultaneously to type. Letters can be typed continuously in accordance to the user's comfort level. The letter is repeated to the user after it registers[12].

Table 2 describes in detail the strengths and weaknesses of the aforementioned literary works carried out by different researchers with respect to the touchscreen interfaces provided by them and the method of text entry for the Blind.

Table 2 Table showing the strengths and weaknesses of previously done work with respect to touchscreen interfaces and mobile-based text entry for the Blind

Approach		Strength	Weakness
Slide Rule[1]	Kane, S.K, et al.	Highlights the point that touchscreen interfaces improve usability for blind users; Incorporates a few audio-based multi-touch interaction techniques	QWERTY keyboard used; no support for larger touch screen displays
QWERTY[3,9]	Noyes, Jan.	Enables blind users to input text in a similar fashion as a sighted user using a simple screen reader	There are a large number of small targets which are difficult to locate for those who are not used to the layout.
MultiTap[6]	Gutowitz, Howard	Layout used is very similar to the ones used in keypad based devices; reduced number of targets on the screen	Number of keystrokes for text input is generally large
NavTouch[4]	Guerreiro, Tiago	Gesture approach-users can navigate horizontally or vertically using vowels as shortcuts to the intended letter	Interaction is limited to directional strokes for every text input; comparatively a slower input method
BrailleType[5]	Oliveira, João, et al.	The layout provides a representation of the Braille cell; easy to explore targets	All the necessary dots for a particular Braille character are to be selected at once.

3 Proposed Work

The aim of developing a smartphone web browser interface for the blind(Fig. 1) is to enable blind users to browse the internet comfortably and to still have all the features of a typical browser, such as bookmarks history, etc., available to them. There are three innovative components of the Ember design that will be discussed: the interface, verbal bookmark search by name and the Ember Braille keypad.

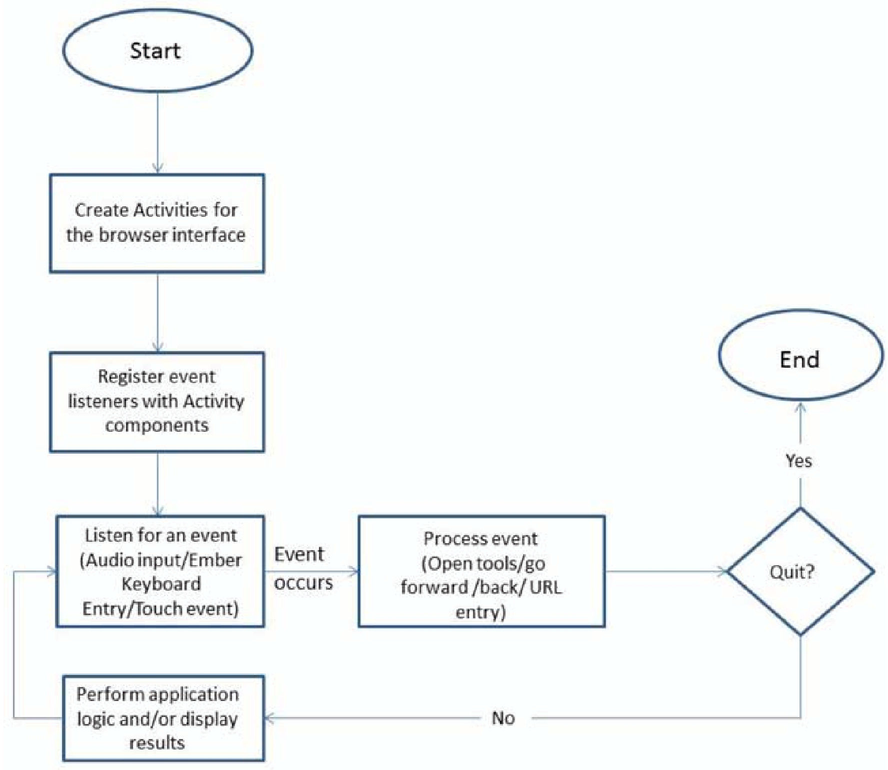

Figure 1. Framework Diagram

3.1 The interface

3.1.1 Layout

The layout of Ember as shown in Figs. 2 and 3, aims to achieve larger target size and a less number of targets[14]. The targets must be easy to locate. There are five large buttons placed along the edges of the phone: forward, backward, tools, audio input and home. The user can use the edges of the phone as bearings. The large centre space in the middle of the layout is the edit box. There is a large area that the user can touch to initiate text entry. This way the user is never forced to pinpoint an exact location on the screen. Although the layout is not comparable aesthetically to mainstream browsers, Ember is much more compatible to the blind user's needs. The user has the option to go back to the home screen from any point in navigation. The layout of the browser is in this format throughout the application so that the user can familiarize with it easily.

Using an assist app like TalkBack on the Android platform, the user can explore the interface without the fear of making a mistake. The name of the screen element is read out to the user when they touch it. Selection is done using double tap and scrolling is done with two fingers. Even menus are announced as they pop up on the screen. The combination of easy to locate target and audio feedback allows the user to adapt to the interface easily and explore with confidence[13].

The targets are placed at the edges of the screen and the text area is the largest section of the interface.

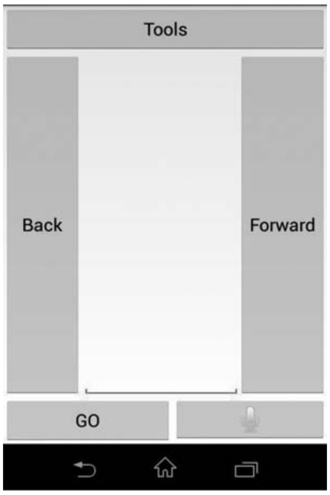

Figure 2. Screenshot of Ember home screen

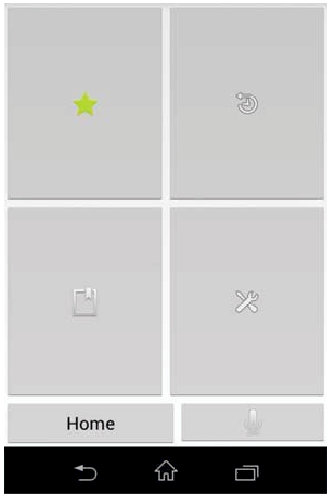

Figure 3. Screenshot of Ember tools screen

3.1.2 Navigation

Navigation is done by two means, by touching the appropriate button or by providing voice commands. On the long press of the audio button, the browser captures the voice command and moves to the corresponding activity on successful processing. This dual method of navigation is designed to provide the user with the most convenient method of interaction according to the situation they are in. For example if the user is in a library then they can use the touch method of navigation but if they are alone at home then the easiest method would be to just use verbal commands.

3.2 Verbal bookmark search by name

A particularly novel feature of Ember is verbal bookmark search by name. When the user chooses to add a page to bookmarks they can name the bookmark. The user presses the audio input button and says the name of the bookmark. The audio input is converted to text and after text to speech processing the interface plays back the name of the bookmark to the user. The user can either choose to accept the result to cancel and record again. When the user wants to access the bookmark again, they just have to give the bookmark name as a verbal command in the bookmarks menu. They can also scroll through their bookmarks as in a traditional web browser.

3.3 The ember braille keypad

3.3.1 Text input

The Ember Braille Keypad integrates the blind user's knowledge of Braille with the requirement of having fewer targets on the touchscreen interface at once[7,8]. The typical QWERTY keypad on a touchscreen phone has an overwhelming number of targets which can be difficult for even the most experienced blind user to use[9]. The buttons are small and it is extremely difficult to type without errors using a trial and error method. The Ember Braille keypad, like the BrailleType keypad, requires the user to map the Braille letter on the touchscreen instead of selecting a button for a particular letter[10]. The screen is divided into six zones so there are only six large targets the user has to handle. Unlike BrailleType, the Braille Keypad uses multi-touch technology to register multiple points of user interaction with the screen at a certain instance of time. This means that the user forms a letter by touching all required zones at once. For this reason there is no timeout mechanism as in BrailleType, reducing the number of errors caused by differences in user competency. The user is provided with audio feedback every time a letter is registered by the system.

3.3.2 Typing on the go

The Ember Braille keypad requires the user to engage at least three fingers on each hand. If the user is using the phone while walking i.e., the phone is not placed on a surface, then the user must hold the phone in a way that he can type comfortably. The phone is held away from the user as shown in Fig. 4. This method ensures that the user is not uncomfortable and holding the phone in an awkward way.

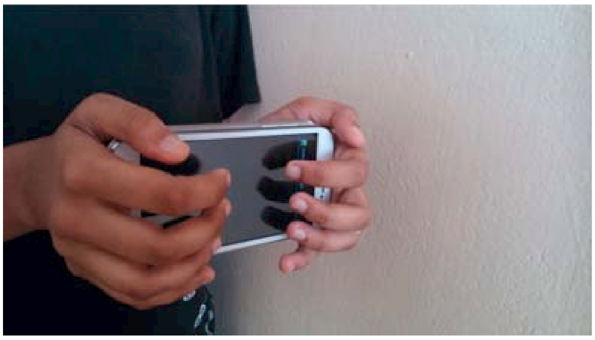

Figure 4. Typing on the go. This method of holding the phone will make typing on the Ember Keypad comfortable even while the phone is not placed on a surface

4. Results and Analysis

4.1 Testing procedure

The Ember interface navigation and Ember Braille keypad were tested by a group of thirteen blind adults who were Braille literates. Ember was tested against a traditional web browser. The Micromax A116 and the Sony Xperia C2104 were used in the testing procedure. To test the ease of navigation, volunteers were instructed to choose the menu, prompt the add bookmark activity, add a new bookmark and then search for the same in the bookmarks library. The volunteers were asked to perform the same navigation test three times for each of the web browsers. The difference in time between each of the three trials for each browser indicates the learning curve while the average time taken by each browser shows the overall complexity that the volunteers encountered.

Table 3 Comparison of navigation time. Navigation time shows the complexity presented by each interface. Ember interface shows less average time

Sl. No of Volunteer	Audio(seconds)		Tactile(seconds)	
	Ember	Normal	Ember	Normal
1	187.65	195.23	140.23	151.67
2	190.25	191.66	135.72	142.53
3	199.56	201.24	142.36	145.93
4	185.37	190.34	138.24	149.25
5	195.9	197.96	139.91	145.73
6	203	208.31	141.11	153.72
7	192.66	192.23	152.71	155.97
8	200.42	206.73	147.84	152.67
9	207.89	209.28	155.71	167.23
10	204.4	202.45	143.75	159.53
11	189.59	192.52	148.89	150.77
12	192.3	193.96	152.6	155.25
13	210.73	209.24	157	161.38
14	209	211.79	139.68	144.24
15	187.34	189.46	148.45	152.76
16	199.7	204.28	152.54	156.23
17	185	187.51	156.71	163.39
18	207	209.43	149	156.93
19	205.63	208.71	150.83	158.34
20	213.52	215.27	158.32	160.27
21	188.75	190.24	144.6	151.62
22	195.92	198.92	159.12	166.92
23	199.6	199.8	150.77	159.37
24	205.45	209.23	162.62	167.67
25	203.62	205.51	153.46	160.52
26	193.68	194.82	151.78	162.45
27	196.35	198.48	149.23	159.28
28	207.73	210.23	155.57	164.96

The Ember Braille Keypad was tested by asking the volunteers to type the letters A C, H Q and R in sequence three times. The time taken by the volunteer to successfully type all five letters was recorded. The letter “A” represents a letter with only one zone interaction, “C” represents two zone interaction, “H” three zone, “R” four zone an “Q” five zone interaction. The same test was repeated with the traditional QWERTY keypad. The comparison between the total times of each trial between both keypads represents the difference in complexity presented by each one.

The Ember Keypad was compared to SmartBraille, designed by tm-laboratory. This is another Android application keypad with Braille support. The approach to Braille mapping is very different between the two applications. SmartBraille is designed for users to enter letters in Braille using a tap or a flick. Its operation is split into three stages from the top down, like a 3 × 2 matrix. If the point to the left is to be filled, then the user must flick left. The user flicks right to fill the point to the right and down to fill both. After each phase the screen is tapped. The volunteers were asked to type a phrase with both keypads. The user's improvement from trial to trial indicates the learning curve associated with either keypad.

4.2 Results

Table 3 shows that twenty-five out of twenty-eight volunteers could on average navigate faster using the Ember audio navigation method than by using the traditional web browser. Twenty-eight out of twenty-eight users could navigate faster using the Ember tactile method of navigation compared to the traditional web browser navigation.

Table 4 depicts how the learning curve for the Ember interface is less steep than the learning curve for the traditional browser. The average trail times for both audio and tactile navigation for the Ember and traditional browser interface show that the time required to complete the navigation test consistently decreased in case of the Ember interface, but was random in the case of the traditional web interface.

Table 4 Comparison of learning curve. The difference in navigation time between each trial indicates the learning curve. The Ember interface shows a decreasing trend whereas normal interface times are random

No. of Trials	Audio(seconds)		Tactile(seconds)	
	Ember	Normal	Ember	Normal
1	210.16	211.49	157.24	159.78
2	208.12	213.96	154.93	160.03
3	205.24	207.25	152.71	161.23
4	202.39	206.44	149.45	158.52
5	201.6	210.13	145.9	162.6
6	197.23	209.61	144.56	160.96
7	194.34	208.19	142.35	164.45

Table 5 shows the average time taken by each volunteer to successfully complete typing the letters A, C, H, Q and R on the Ember keypad and the QWERTY keypad. It is seen that the time taken to type using the QWERTY keypad is significantly longer in the case of all volunteers.

4.3 Analysis

The consistent downward slope presented by the Ember interface in Figs. 5 and 6 clearly shows that the volunteer's performance improves significantly with every trial. The traditional web browser shows random results with the slope showing an upward trend and downward trend over the three trials. The Ember curve in the tactile navigation map has a consistent downward slope, but the slope drastically increases from trial two to trial three in the audio method of navigation graph. This indicates that audio method of navigation is even easier to get accustomed to than the tactile method.

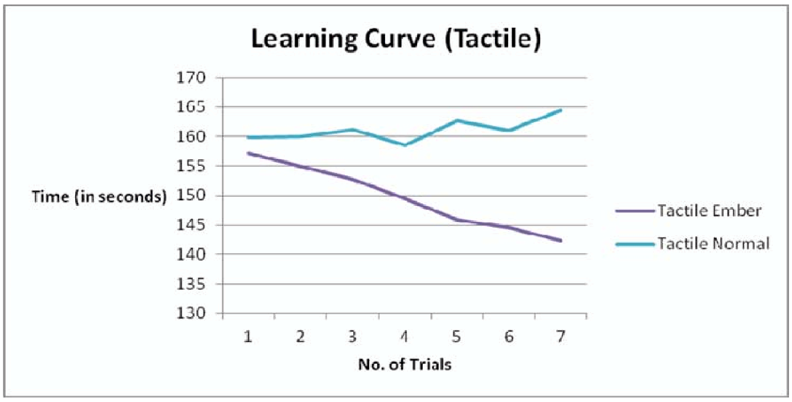

Figure 5. Navigation Learning curve (Ember v/s normal browser – Tactile)

Table 5 Ember keypad test for complexity

Sl. No of Volunteer	Average Time (seconds)	
	Ember	Normal
1	30	59
2	26.67	49
3	33.33	51
4	39	48.33
5	27	53
6	21.33	45
7	26	53.67
8	25	41
9	32	47
10	37	46.33
11	45	55.67
12	31	52.33
13	28	41
14	26	40
15	33.33	49.33
16	30	48
17	28	45
18	45.33	45
19	32.667	40
20	33	42.33
21	27	39.5
22	29	52
23	31	48.33
24	30	45.67
25	28	41
26	21.33	45
27	32	37.5
28	29	40

Figures 7 and 8 show that the time taken to perform the navigation test was almost always longer in the case of traditional web browser compared to the Ember interface. It is also seen that the time taken to complete the audio navigation test is more than the time taken to complete the tactile navigation test. Tactile navigation is faster than verbal command navigation.

Figure 9 shows that the time taken to complete typing the sequence of letters is significantly less in the case of Ember Keypad compared to the QWERTY keypad.

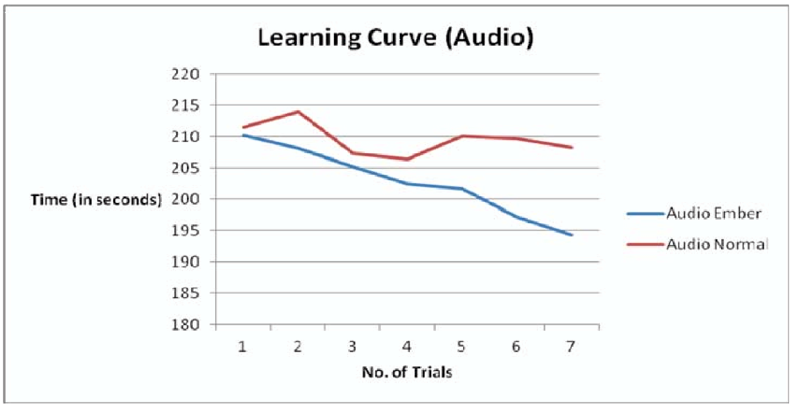

Figure 6. Navigation Learning curve (Ember v/s normal browser – Audio)

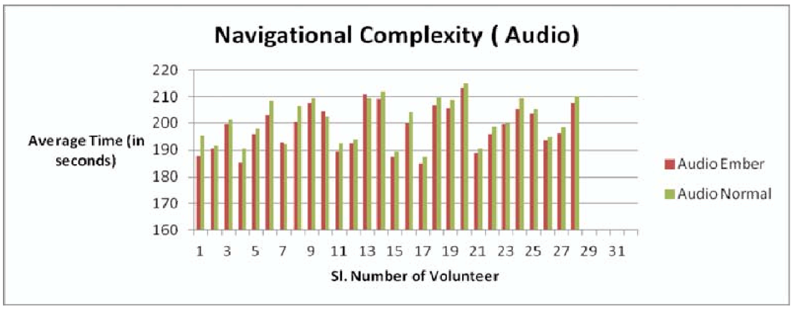

Figure 7. Navigation Complexity evaluation (Ember v/s normal browser – Audio)

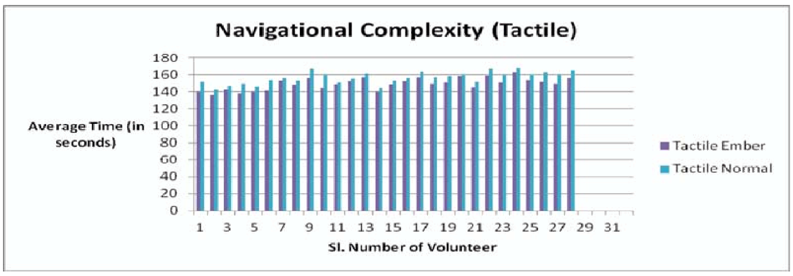

Figure 8. Navigation Complexity evaluation (Ember v/s normal browser – Tactile)

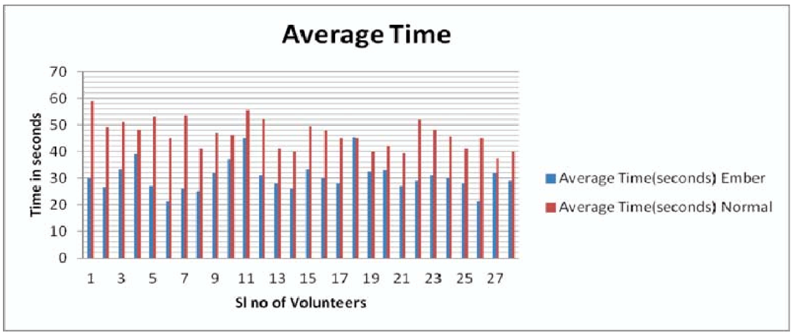

Figure 9. Complexity evaluation (Ember v/s normal browser – Ember keypad)

4.4 Discussion

After testing and analyzing it was found that the volunteers found it easier to use the Ember interface compared to a traditional web browser. The analysis showed that with the use of the Ember keyboard it took them lesser time to input text. The Ember interface was found to be more accessible than the traditional smartphone web browser in terms of navigation and text input. The navigation through Ember was made easier with the use of speech input and audio feedback at every stage. The complexity of navigation (for both audio and tactile navigation) as well as the learning curve was smoother in the Ember interface compared to the traditional web browser.

The volunteers required a period of time to get accustomed to the Ember interface. It was seen that the exploration phase was relatively simple and quick when compared to the traditional web browser. A few volunteers commented that the traditional web browser was intimidating and would require a lot of practice to master. When asked which feature they appreciated most about the Ember interface, the volunteers concentrated on the ease they felt with locating large and the constant audio feedback.

It was seen that the volunteers faced difficulty with the QWERTY keypad. They found the targets too small and locating individual letters cumbersome. On average the volunteers took about one to two minutes to get accustomed to the Ember Keypad. They were comfortable with the Braille based entry. The Ember Keypad still faced errors mainly due to asynchronous selection of targets. If the volunteer intends to type a certain letter, then he must select all the required targets at the same time. The volunteers faced trouble typing the correct letter if they show hesitation while in the process of typing the letter. This delay caused the keypad toe register the wrong letter.

Testing after phase one of development provided constructive user feedback. One of the points raised was that it was difficult for the volunteer to locate the six regions of the screen and place their fingers accordingly. To remediate this problem a customized screen guard was introduced. A screen guard is a plastic covering having one adhesive side that is applied on to the phone screen to protect the screen from scratches. The screen guard had six small protrusions each indicating the center of an area. After the screen guard was used it was much easier for the volunteers to find their bearings. The question of deployment was raised. Though the customized screen guard was an elegant and simple solution to the problem, the volunteers asked questions about where such a screen guard could be obtained. Since the screen guard was made by the Ember team exclusively for testing the solution, there are no other screen guards of this type available. If the Ember keypad is to be used in the future, the customized screen guards could be made on a large scale. This solution was concluded to be a feasible and effective solution.

After testing the Ember keypad against SmartBraille, volunteers found that the Ember keypad was easier to learn and faster to type sentences because each letter in Ember was not a three phase process. Volunteers also found that Ember has a direct mapping to Braille whereas SmartBraille's is more abstract. A few volunteers found that after the initial learning curve, SmartBraille was easier to use because they did not have to map their fingers to the various regions of the Ember Keypad screen. The user only has to flick and tap the screen in the case of SmartBraille. This study signified that the Ember Keypad is on par with an established application and presents a few advantages compared to it as well.

The feedback given by the volunteers was overwhelmingly positive one of the volunteers stated that it was very easy to understand and would be extremely useful with a little practice. Furthermore, one volunteer stated that the Ember Keypad would be an enjoyable and easy way to learn braille. The ophthalmologists consult during the testing process understood Ember's design easily. They found the principles with which the interface was designed to be clear.

5 Conclusion

The goal of this research endeavor was to make technology and services practical and effective for blind users. With an average of twenty-five out of twenty-eight volunteers finding audio navigation with the Ember interface significantly better the traditional web browser, the Ember interface confidently presents a new platform for the blind. It also assures an optimistic experience because the Ember interface does not compromise on any features when compared to the traditional web browser. The Ember interface consists of easy-to-locate, large targets consistently placed throughout the application enhancing, usability greatly. The Ember keypad allows the blind to type using their knowledge of Braille and thus facilitates faster and accurate text input. The user experience with the Ember Keypad was further improved by using a screen guard with six small protrusions in order to locate the six regions on the screen for Braille based text entry.

6 Future Work

Ember is still in its nascent stages and can be improved upon in a number of ways. More features such as gestures can be added to the interface to make it more accessible, gestures to begin searching, open ember keypad, scroll lists easily. A few smartphones could also incorporate air gestures for better navigation between pages. The comparison of the Ember Keypad to other braille input methods would be a more appropriate comparison to test its usability further. As one of the volunteers had suggested, creating an Ember Keypad exclusively for teaching braille would be a rewarding extension of the project.

Ember as of now does not include a web reader, a read aloud application that would read the results of a search or listings aloud, including such a feature would make Ember more handy and easy to use.

The Ember keypad has been designed for English braille, other languages can also be incorporated effortlessly in future with features that could use gestures to input text.

Acknowledgement

The completion of this work would not have been possible without the support and cooperation of the Roman and Catherine Lobo School for the Blind, Mangalore, Mobility Training Centre for the blind, Attavar, Mangalore, Dr. Govindraj at Prathibha Eye Hospital, Mitra Jyoti, Bangalore, a charitable trust that aims to support the blind.

References

[1] Kane SK, Bigham JP, Wobbrock JO. Slide Rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. Proc. of ASSETS '08. New York: ACM Press. 2008. 73–80.

[2] Oliveira J, et al. Blind people and mobile touch-based text-entry: acknowledging the need for different flavors. Proc. of the 13th International ACM SIGACCES Conference on Computers and Accessibility. ACM. 2011.

[3] Noyes J. The QWERTY keyboard: A review. International Journal of Man-Machine Studies, 1983, 18.3: 265–281.

[4] Guerreiro T, et al. Mobile text-entry models for people with disabilities. Proc. of the 15th European Conference on Cognitive Ergonomics: The Ergonomics of Cool Interaction. ACM, 2008.

[5] Oliveira J, et al. BrailleType: unleashing braille over touch screen mobile phones. Human-Computer Interaction – INTERACT 2011. Springer Berlin Heidelberg, 2011. 100–107.

[6] Gutowitz H. Method and apparatus for improved multi-tap text input. U.S. Patent. 6,219,731. 17 Apr. 2001.

[7] Sergio M, Bernareggi C, Belotti M. TypeInBraille: a braille-based typing application for touchscreen devices. The Proc. of the 13^{th} International ACM SIGACCESS Conference on Computers and Accessibility. ACM. 2011.

[8] Sergio M, Bernareggi C, Belotti M. Typeinbraille: quick eyes-free typing on smartphones. Computers Helping People with Special Needs. Springer Berlin Heidelberg, 2012. 615–622.

[9] Clawson J, Lyons K, Starner T, Clarkson E. The impacts of limited visual feedback on mobile text entry for the twiddler and mini-QWERTY keyboards. Proc. Ninth IEEE International Symposium on Wearable Computers, IEEE Computer Society. 170–177. 2005.

[10] Jussi R, et al. Methods for presenting braille characters on a mobile device with a touchscreen and tactile feedback. IEEE Trans.s on Haptics, 2009, 2.1: 28–39.

[11] McGookin D, Brewster S, Jiang WW. Investigating touchscreen accessibility for people with visual impairments. Proc. of the 5th Nordic Conference on Human-computer Interaction: Building Bridges. ACM. 2008.

[12] Vanderheiden GC. Use of audio-haptic interface techniques to allow nonvisual access to touchscreen appliances. Human Factors and Ergonomics Society Annual Meeting Proceedings, 1996, 40: 1266

[13] Yfantidis G, Evreinov G. Adaptive blind interaction technique for touchscreens. Universal Access in the Information Society, 2006, 4(4): 328–337.