One handed text input in Devanagari

एक हाथ द्वारा देवनागरी में टेक्स्ट इनपुट

Interaction design project II

By-

Prashant Sachan
0816440022

Guided by-

Prof. Anirudha Joshi
IIT Bombay

November 2013

Industrial Design Centre,
Indian Institute of Technology Bombay
Mumbai, India

This Interaction Design degree project, titled **‘One-handed text input in Devanagari’** is done in partial fulfillment of the requirement for the degree of Master of Design in Interaction Design at Industrial Design Centre, IIT Bombay.

Duration: August to November, 2013

Approval sheet

Interaction Design Degree Project entitled
'One handed text input in Devanagari'

By: Prashant Sachan (126330004)

is approved in the partial fulfillment of the requirements of the degree of Masters of Design in Interaction Design, December 2013

Guide:

Chairman:

External Examiner:

Internal Examiner:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all the principles of academic honesty and Integrity and have not misrepresented or fabricated or falsified any idea/data/fact/ source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

Prashant Sachan
Interaction Design, IDC, IIT Bombay
Roll no. 126330004

Mumbai, India

18th November 2013

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my guide, Prof. Anirudha Joshi for their exemplary guidance, monitoring and constant encouragement throughout the course of my project.

I also take this opportunity to express my thankfulness towards Prof. Girish Dalvi and Prof. Pramod Khambete for the valuable information and guidance, which helped me to carry the project in a better way.

I would like to thank Deepak Malani (Research scholar, IIT B) for helping me in troubleshooting the code for the prototype. I would also like to thank Sajal Nagwanshi and Abhinav Kumar (fellow IDCians) for their support while I was working on the prototype.

Lastly, I thank almighty, my parents and my brother for their constant encouragement and blessings.

Table of contents

1

Abstract

Efficient physical devices, specially designed for one-handed text input in Devanagari aren't available. This projects aims at designing an efficient one-handed text input method for Devanagari such that it facilitates the typing process with just one hand. The project contributes new methods to input Devanagari text, based on the structure of the script. The designs underwent through cycles of design, prototyping and evaluation.

The final prototyped and evaluated concept called 'Melo' is inspired from 'melodica' – a type of musical keyboard. In Melo, characters are entered through simultaneous multiple key presses, using key combinations that follow a pattern based on the structure of the script.

Two types of evaluations were conducted on the design- One was Theoretical evaluation, in which overall effort was measured and compared with InScript and Half-qwerty keyboard (both were used for one-handed input in Devanagari) for 3 set of words with different level of complexity; and other was Empirical evaluation, in which a short term test and long term test were conducted for the final design and InScript Keyboard (for one-handed use). The objective was to compare learnability, typing speed and accuracy for the two keyboards.

Melo keyboard was found intuitive and easy to learn by first-time users. The finger travel and no. of key presses on Melo were similar to Half-qwerty, Inscript had least no. of key presses and maximum finger travel. But, the finger gestures while typing on Melo were user-friendlier and the finger load was distributed primarily on the fingers with higher strength and pointing accuracy as compared to the other two. Based on the data from Long term Study, Melo was doing better than InScript but, the difference were not significant enough to draw conclusions about the typing speed and accuracy.

2

Introduction

'The primary job of a clerk at a bank was to type in Devanagari. Unfortunately, he damaged his one hand in an accident and because of that, his accuracy and speed of typing was badly affected.'

The initial motivation to start with the project was to facilitate the job of typing for the people who damage their one hand, i.e.. To design for One-Handedness.

The aim was that the design can be used as a standard desktop keyboard that allows the user, with no use or limited/restricted use of one hand, to type in a natural manner and in an easy way.

Also, an efficient one-handed text input device will act as a facilitator in the process of healing of the affected hand in cases of temporary damages on one-hand, Carpal Tunnel Syndrome and other typing related injuries, as such injuries don't generally occur in both the hands at a time. [1]

Such input methods will not only benefit the users with hand injuries or limited one-hand use, but also it will benefit a normal user with both hands in perfectly fine state. If the one hand is free while typing, enormous possibilities can be explored [1] [2].
For example -

- Other input devices such as mouse or a drawing tablet can be used with the free hand. Also, a lot of actions such as scroll, select text or editing in type involves moving the hand back and forth between keyboard and mouse, one-handed keyboard will be quite useful in this scenario.
- If one hand is free, the documents can be held with it and typing can be done with the other hand.

While studying the available one-handed text input methods, it was found there are various physical keyboards available for one hand typing in Latin script, for example- frog pad, half qwerty keyboard etc.; Where as nothing of that kind which is specially designed for one-handed text input is available for text input in Devanagari. If an efficient one handed text input method is available for Devanagari, then the user can achieve a decent typing speed with just one hand instead of using both the hands.

This also grew as a strong motivation as successful execution of project will contribute new methods of one-handed text input in Devanagari that wasn't really worked upon in the past.

The target user for this design project is a normal user who can read and write in Devanagari. The goal is to enable such users to input with just one hand, also the design should be easy to learn and use after basic training and product demonstration/walk through.

Design process

The following process was followed-

- Study of structure of Devanagari
  It also involved understanding the conceptual model of the script for the users and observing them while they learn to type on an Indic script keyboard.
- Study of various text input methods and existing one-handed text input devices
- Formulating product brief and design goals
- Ideation + Conceptualization
- Prototyping
- Evaluation of the concepts

3

Devanagari is part of the Brahmic family of scripts of India, Nepal, Tibet, and South-East Asia. It is an abugida[1] alphabet, written from left to right. It is used to write Hindi, Marathi, Nepali along with Awadhi, Bhojpuri, Konkani, Maithali, Santhali etc. [3]

The letter order of Devanagari, like nearly all Brahmic scripts, is based on phonetic principles that consider both the manner and place of articulation of the consonants and vowels they represent. This arrangement is usually referred to as the varnamala (garland of letters). [4]

1 "An abugida is a segmental writing system in which consonant–vowel sequences

3.1 **Structure of letters in Devanagari**

अ आ इ ई उ ऊ
ए ऐ ओ औ अं अः

क ख ग घ ङ
च छ ज झ ञ
ट ठ ड ढ ण
त थ द ध न
प फ ब भ म
य र ल व
श ष स ह

क्ष त्र ज्ञ श्र

Figure 1: Devanagari Varnamala

Figure 1 shows Devanagari Varnamala, which shows most frequently used characters, which includes 12 vowels (अ आ इ ई उ ऊ ए ऐ ओ औ अं अः), 33 consonants (क ख ग ...ष ह) and 4 conjuncts (क्ष त्र ज्ञ श्र). The last eight of the 33 consonants are also known as semi-vowels. [5] Because the letter order is phonetic in Devanagari, the Varnamala is learnt like a song, it's sung and learnt. The structure has small phonetic groups of characters, for e.g.. (क ख ग घ ङ), (च छ ज झ ञ) etc.

The image below (Fig.2) shows the arrangement of vowels based on phonetic principles. [3]

	Independent form	As diacritic with प	Independent form	As diacritic with प
kaṇṭhya **(Guttural)**	अ	प	आ	पा
tālavya **(Palatal)**	इ	पि	ई	पी
oṣṭhya **(Labial)**	उ	पु	ऊ	पू
mūrdhanya **(Retroflex)**	ऋ	पृ	ॠ	पॄ
dantya **(Dental)**	ऌ	पॢ	ॡ	पॣ
kaṇṭhatālavya **(Palato-Guttural)**	ए	पे	ऐ	पै
kaṇṭhoṣṭhya **(Labio-Guttural)**	ओ	पो	औ	पौ

Figure 2: Arrangement of vowels based on phonetic principles.
Source: http://en.wikipedia.org/wiki/Devanagari [3].

The image below (Fig.3) shows the arrangement of consonants based on phonetic principles. [3]

	***sparśa* (Plosive)**				***anunāsika* (Nasal)**	***antastha* (Approximant)**	***ūṣma/saṃghaṣhrī* (Fricative)**	
Voicing →	*aghoṣa*		*ghoṣa*				*aghoṣa*	*ghoṣa*
Aspiration →	*alpaprāṇa*	*mahāprāṇa*	*alpaprāṇa*	*mahāprāṇa*	*alpaprāṇa*		*mahāprāṇa*	
***kaṇṭhya* (Guttural)**	क	ख	ग	घ	ङ			ह
***tālavya* (Palatal)**	च	छ	ज	झ	ञ	य	श	
***mūrdhanya* (Retroflex)**	ट	ठ	ड	ढ	ण	र	ष	
***dantya* (Dental)**	त	थ	द	ध	न	ल	स	
***oṣṭhya* (Labial)**	प	फ	ब	भ	म	व		

Figure 3: Arrangement of consonants based on phonetic principles.
Source: http://en.wikipedia.org/wiki/Devanagari [3].

3.2 **Classification of symbols/letters in Devanagari**

Vowels:	अ आ इ ई उ ऊ ऋ ए ऐ ओ औ अं आः
Vowel Modifiers:	◌ा ◌ि ◌ी ◌ु ◌ू ◌ृ ◌े ◌ै ◌ो ◌ौ ◌ं ◌ः
Consonants:	क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न प फ ब भ म य र ल व श ष स ह ळ
Popular Conjunct:	क्ष श्र ज्ञ त्र
Diacritic Marks:	◌ँ ◌ं . . । ऽ ◌़

Figure 4: Classifications of symbols in Devanagari script. Source: [6]

The symbols shown in vowels (V) (Fig.4) are the standalone letters for vowel sounds. Vowels are pronounced independently and they are used to help in the pronunciation of full consonants.

The consonant (C) (Fig.4) symbols are shown above, they have an inherent ‘ अ ’ (a) vowel sound. And they are known as full consonants. In Varnamala, they are represented as full consonants. The diacritical mark ‘ ◌् ’ (halant) is added to a consonant to remove that ‘ अ ’ (a) vowel sound in it. And that is the pure sound of a consonant. The consonants with the diacritical mark ‘ ◌् ’ (halant) are pure consonants.
क् + अ = क

Vowel Modifiers, also called ‘Matras’, represent the vowel sound in the consonant in the combination of a consonant and a vowel. Vowel modifier symbols cannot be used as a standalone letter for a vowel sound. It can work only as an attachment to a consonant. Also, in the script, there is no special matra for ‘ अ ’ because consonant symbols have an inherent ‘ अ ’ sound.

The diacritical symbols –
‘ ◌ं ’ (anuswar) is a special symbol, when it is attached to consonant or vowel; it indicates its nasalization.
‘ ◌ः ’ (visarg) similarly is used indicate ‘h’ sound after a consonant or a vowel to which it is attached.
‘ ◌़ ’ (nukta) if attached to a consonant, it represents its Persian pronunciation and
‘ ◌ँ ’ (Chandra bindu) when added to a consonant denotes nasalization of the preceding vowel.

Conjuncts (fig: 4), some symbols are assigned to combination of consonants using diacritical mark ‘ ◌् ‘ (halant).

For example: The popular conjuncts- क्ष, त्र, ज्ञ, श्र are combinations of consonants क् ष, त् र, ज् ञ, श् र respectively i.e.. These can be represented in either of the two ways. [7]

3.3 Challenges in text-input in Devanagari

The issues related to input in Devanagari have been extensively mentioned and identified in the works related to text input in Indic scripts. [5] [6] [7] [8]
The challenges are-

- Character formation in Devanagari involves multiple keystrokes
- Difference in cognitive styles of writing and typing (speaking)
- Large number of characters

3.3.1 Character formation in Devanagari [5]:

To form a character, one or more letters combine together. To form a word, one or more such characters come together. There are a lot of combinations for Consonants and vowels in which they can come together to form characters. Understanding of such combinations is required for error-free text input in Devanagari. Also, the method to achieve them on an Inscript[2] keyboard involves multiple key presses and a user is required to develop a conceptual model of the same. Different combination patterns and methods are described below:

C+V: Each character ends with a vowel in Devanagari. A simple character consists of a combination of a consonant and a vowel (C+V).

[2] "**InScript** is the standard keyboard layout for Devanagari using a standard 104 or 105 key layout. It is standardized by the Government of India for text input in Devanagari." [25]

For demonstration, the word (paanee) consists of two characters (pa) and (nee).
पानी = पा + नी
paanee = paa + nee

The first character (pa) is combination of the consonant (p) and the vowel (aa), and the other character (nee) is formed by combination of consonant (n) and the vowel (ee).
पा + नी = प् + आ + न् + ई
paa + nee = p+aa + n+ee

In Inscript, the vowel (a) is added by default when a consonant key is pressed. For example, if 'k' key is pressed, 'क' is printed. 'अ' is added by default to 'क्'.
If another vowel is desired, the user needs to explicitly press the corresponding vowel key. To type the character 'का' (kaa), press 'क' followed by Matra ' ा '.
To type 'का' (kaa) on InScript:
का = क + ा

C+C+V: A more complex character is in which two or more consonants combined with a vowel or a diacritic mark (also called conjunct). For example, the word (prem) consists of two characters, the first character (pre) is made up of consonant (p), conjunct (r) and vowel (e), and the other character is just full consonant (m).
प्रेम = प्रे + म = प + ऱ् + ए (= प + र + ् + ए)+ म् + अ
prem = pre + m = p + r + e + m + a

To type it in Inscript, user presses the first consonant key (p), followed by the second consonant key (r), followed by diacritical mark ' ् ' (halant), and followed by the vowel key (e). To get the character (m), consonant key (m) is pressed.

C+V+V: Another combination is Consonant combines with two vowels. For example in the word (paande), it has two characters, one is (paan) and other is (de).
Conceptually,
पांडे = पां + डे = प् + आ + अं + ड् + ए
paande= paan + de = p + aa + n + d + e

To type it in Inscript, the first character is input by pressing the consonant key (p) followed by the two vowel keys (aa) and (an), then other character is input by pressing consonant (d) followed by vowel (e).

3.3.2 **Difference in cognitive Styles of Writing and Typing** [5]:

In some characters, the visual sequence in which user writes the corresponding glyphs is different from the phonetic sequence of consonants and vowels. [5]
For example, for the character (pi), users write the glyph for the vowel (i) before the glyph for the consonant (p), but pronounce the (i) after the (p).
पि = प् + इ
pi = p + i

In another example, the word (arth), users write the glyph for the consonant (r) after the consonant (th) where as the sequence of pronunciation of the character is (rth). Users think in the visual sequence of the glyphs when they write.

अर्थ = अ + र्थ = अ + र् + थ् + अ
arth = a + rth = a + r + th + a

Computer storage formats such as Unicode and ISCII (Indian Standard Code for Information Interchange) have standardized the phonetic sequence of consonants and vowels for typing.

These differences between the phonetic and visual sequence don't apply to all the characters and aren't frequently occurring, but these differences between typing sequence and writing sequence are quite significant to trouble the beginners/novices and it requires learning.

3.3.3 **Large number of Characters** [5]:

As discussed in section 3.3.1 on 'Character formation in Devanagari', combinations of (C+V), (C+C+V)... (C+C+…V)…can be used to generate infinite number of Devanagari characters theoretically. Actually, a smaller subset of characters is currently used in Devanagari. But even this small subset of characters is quite large in number that makes it nearly impossible to design a practical keyboard where one key corresponds to one character.

For example, in a standard QWERTY keyboard, one key corresponds to one character and with that number of keys (26) it's practically impossible to accommodate even the 53 base letters [5] of Devanagari where one key corresponds to one Unicode character.

Increasing the number of keys is not a solution, as that will increase the size of the keyboard, which will lead to a lot of hand movement. Also, more number of keys cannot be accommodated in the same keyboard size (of standard Qwerty keyboard) as that will lead to smaller key size that will adversely affect accuracy (as per Fitts's law).

The key statement of Fitts's Law is that the time required to move a pointing device to a target is a function of the distance to the target and its size. The deductions that can be made from law can be summed up as: the closer and larger a target, the faster it is to click on that target. [9] [10]

Thus, it becomes necessary to map more than one character on one key. Existing Devanagari text input solutions adds to significant cognitive load on the users as available methods handle the challenge of large number of characters by force-fitting some base letters on the shifted positions. [5] This leads to multiple keystrokes for character formation in Devanagari.

3.4 **Challenges with one-handed text input in Devanagari**:

The size of the keyboard is an important factor when it comes to one-handed input. Ideally, a keyboard designed for one-handed input should lead to minimum or zero hand movement. So as to achieve that, the size of the keyboard should be comparable to the size of palm of a normal adult user. [2] [11]

So, for a smaller keyboard size (as compared to existing two-handed input InScript keyboard), the major challenge is to establish an optimum balance between the number of keys and the size of key. The size of the key is an important factor as well, because depending on the key size, there will be a trade-off between speed and accuracy. [10]

So considering the challenges involved in text-input in Devanagari (section 3.3) and the constraints involved in designing a keyboard for one-handed input; deciding upon size of the keyboard, number of keys and key size, and thus mapping large number of characters on them is even more intense and complicated process.

Even the existing hardware based Devanagari text input solutions (for normal use) add to significant cognitive load on the users. In case of one-handed keyboard in Devanagari that requires mapping larger number of characters on a key/ key combination, the users will be required to develop complex conceptual model for using it effectively, and that makes one-handed input systems harder to learn.

4

Study of various text input methods

4.1 Popular keyboard layouts

4.1.1 Qwerty Layout

It is the most widely used layout today; the layout was devised and created in the early 1870s by Christopher Latham Sholes for a typewriter. The first model that was designed, was a piano-like keyboard with two rows of characters arranged alphabetically:

```
- 3 5 7 9 N O P Q R S T U V W X Y Z
 2 4 6 8 . A B C D E F G H I J K L M
```

Figure 5: Alphabetical layout
Source: [12]

The alphabetical layout (Figure 5) was susceptible to jamming of mechanical columns/bars in the typewriter. In further iterations, rearrangements were done with the aim to place commonly used letter-pairs (for example.. ‘i s’ , ‘a n’ etc.) so that their type bars were not neighboring, avoiding jams.

As a result, Qwerty layout did a better job of balancing the keys between fingers. The overall score for Qwerty is some 12% better than that of Alphabetical order based layout. [13]

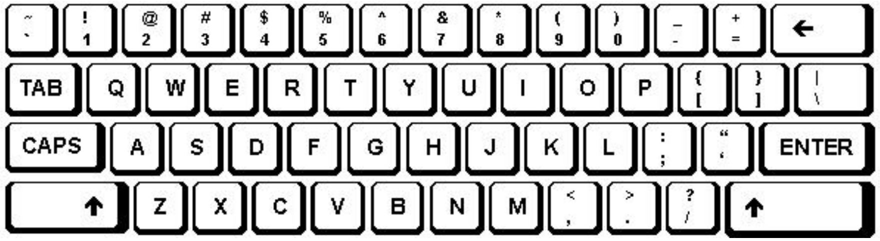


Figure 6: Modern day Qwerty layout
Source: http://www.microsoft.com/enable/images/products/kbqwerty.gif [14]

4.1.2 **Dvorak Layout**

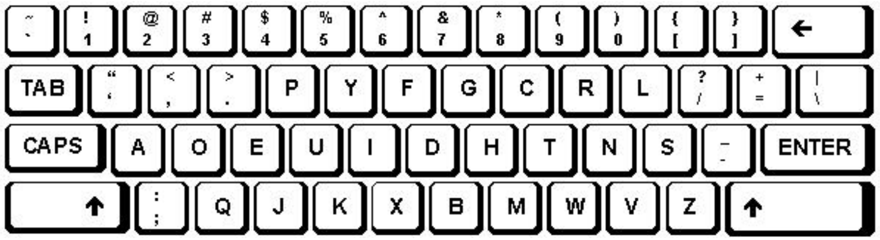


Figure 6a: Dvorak layout
Source: http://www.microsoft.com/enable/images/products/kbdvorak.gif [14]

Dvorak layout was designed to replace Qwerty layout in the first half of the 20th century, and it was optimized for touch-typing[3]. Dvorak layout uses less finger travel, which has led to high typing speed, and less error compared to the standard Qwerty keyboard. [13] Less finger travel has also helped in reducing repetitive strain injuries. [14]

The commonly identified problems with Qwerty were that the finger load was there on fingers of less dominant hand and weaker fingers, also finger travel was quite significant in Qwerty layout. [13,12] Dvorak worked on improving such problems in his design. The layout he created adheres to these principles: [15]

- Letters should be typed by alternating between hands (which makes typing more rhythmic, increases speed, reduces error, and reduces fatigue). On the Dvorak, vowels are all on the left home row, the most used symbols are on the left, while the most used consonants are on the right.
- For maximum speed and efficiency, the most common letters and digraphs should be the easiest to type. This means that they should be on the home row, which is where the fingers rest, and under the strongest fingers (Thus, about 70% of keyboard strokes on the Dvorak Simplified Keyboard are done on the home row and only 22% and 8% on the top and bottom rows respectively).
- The least common letters should be on the bottom row, which is the hardest row to reach.

[3] Type using all one's fingers and without looking at the keys.

4.2 Study of one-handed text input Keyboard layouts

4.2.1 Lewis keyboard

The keys in this keyboard are alphabetically arranged; it is designed for single finger text entry. It gives better text entry rate than QWERTY for one-at-a-time character selection. The user model for typing with 10 fingers is clearly inappropriate for evaluating layouts for typing with one finger. [8]

A	B	C	D	E	
F	G	H	I	J	
K	L	M	N	O	
P	Q	R	S	T	
U	V	W	X	Y	Z

Figure 7: Lewis Layout
Source: [8]

4.2.2 FITALY keyboard

It is one-Finger Keyboard in which the letters are arranged based on frequency. It is optimized for entry with a single finger or with a pen. It has two space bars in the layout, the frequently used characters like E, T, A are placed close to the space bar. The average hand movement is found lower as compared to the Lewis Keyboard, leading to a better text entry speed in long-term use. [8]

q	f	u	m	c	k	z
space		o	t	h	space	
b	s	r	e	a	w	x
space		i	n	d	space	
j	p	v	g	l	y	

Figure 8: Fitaly layout
Source: [8]

4.2.3 Half-Qwerty keyboard

The Half-QWERTY keyboard (Fig.9) is based on a standard QWERTY layout but only a half of the keys are accessible at a time. When a key is pressed, the top-left character is entered. If the SPACE key is held down while tapping on a key, the bottom-right character that is entered. Here, half the characters (the bottom-right characters) are mapped to two keys. [2]

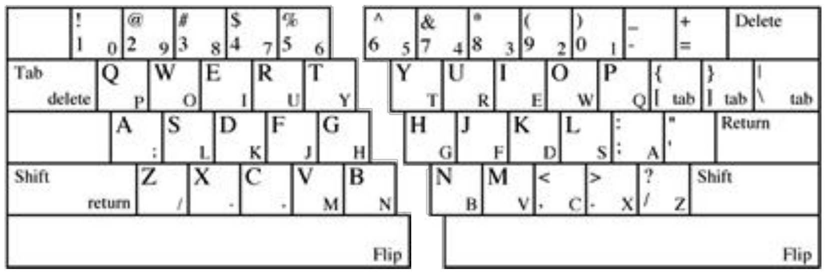


Figure 9: Half-Qwerty layout
Source: [2]

So, this will increase the effort that a user has to put, for e.g.. In RIGHT HANDED Half-Qwerty keyboard for inputting all the bottom-right characters on the keys, the space bar has to be held. This will lead to more errors as the frequently used characters are also mapped in that bottom right section. Also, this will enhance the cognitive load on the users.

4.2.4 **Chorded keyboard**

It allows the user to enter characters or commands formed by pressing several keys together, like playing a "chord" on a piano. It is a one handed text input device that is available for Latin script. As the number of keys is very less, so combinations of keys are pressed simultaneously (multiple key press) to enter a desired character. [16] [11]

4.2.4a **Microwriter**

Here, a Microwriter is discussed (Fig: 10). The 23 cm × 12 cm × 5 cm hand-held device comprises a six-button chording keyboard, a single line LCD display, an 8 bit CDP1802 microprocessor and a complete word processing software in ROM. [16]

Figure 10: Microwriter

Source: [16]

The keyboard uses one button for each finger and two for the thumb of the user's right hand (Figure 11). The five buttons are pressed simultaneously in different combinations to input the letters of the roman script, the other thumb button is used to switch modes, for example- for numerical input, for entering punctuations, symbols etc. [11]

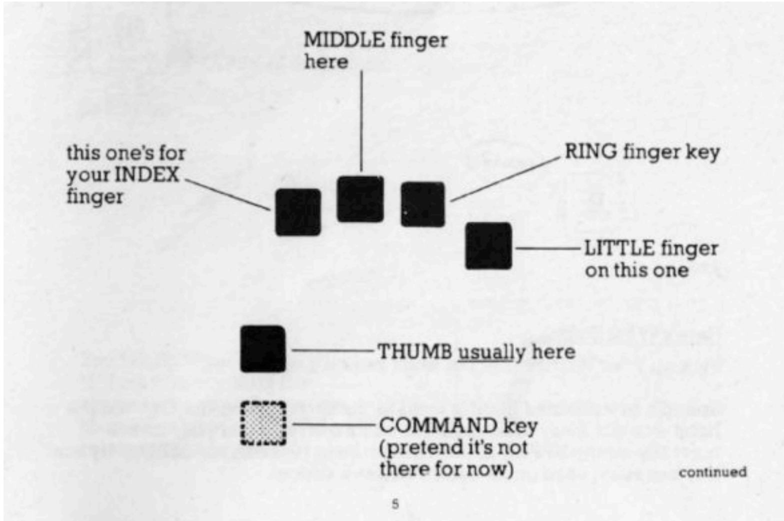


Figure 11: Finger positions and key functions on Microwriter
Source: http://www.computinghistory.org.uk/det/5798/Microwriter-new-users-guide/ [17]

The various key combinations that were assigned for various letters were based on a pattern, which was based on the form of the letters (fig.12) (fig.13) [17]

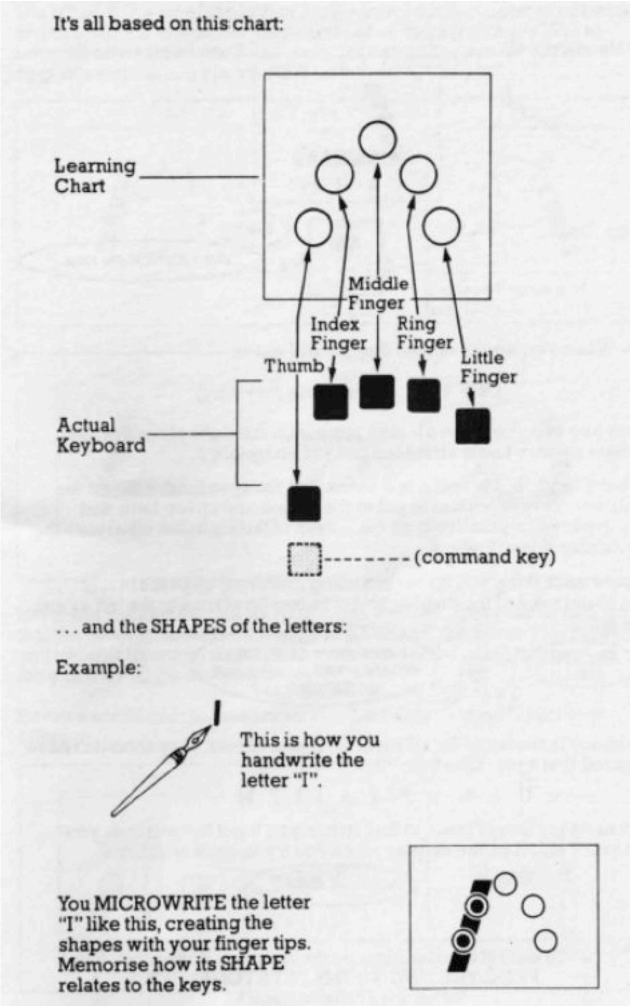


Figure 12: Key combinations for various letters
Source: http://www.computinghistory.org.uk/det/5798/Microwriter-new-users-guide/ [17]

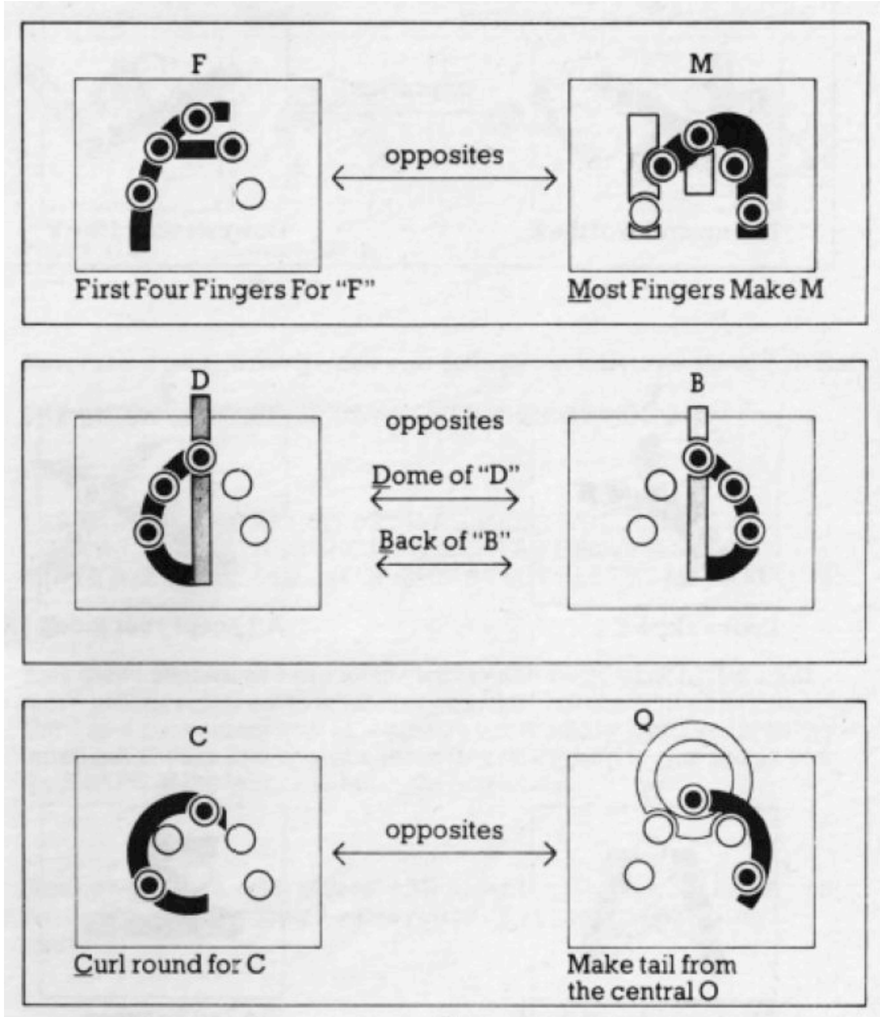


Figure 13: Key combinations for various letters
Source: http://www.computinghistory.org.uk/det/5798/Microwriter-new-users-guide/ [17]

4.2.4b **Cykey**

Microwriter was further designed to cater to both left and right-handed users. It is called Cykey (Fig. 14); it features 9 keys, grouped into three sets of three keys. The functioning is similar to that of the Microwriter.

For example, for right hand use, the thumb is positioned over the left-most button on the bottom and the one over it on the left. While the other four fingers are positioned over the next four adjacent keys. Similarly, for the left hand, little finger, ring finger and middle finger are positioned over the set of three buttons in the middle respectively starting from the left hand side. The thumb is positioned over the right most buttons on the bottom and the one over it. [11] Key combinations for Microwriter (Fig. 12) were similar to Cykey-codes (Fig. 15)

Figure 14: Cykey
Source: http://www.cykey.co.uk/ [11]

Figure 15: Cykey-codes for key combinations for letters
Source: http://www.cykey.co.uk/ [11]

Learning the chords for the basic letters and numbers is facilitated by a set of flash cards that show simple mnemonics for each character (fig. 15). After practice, it is possible to do faster text entry with the Microwriter than with a conventional keyboard. [16]

Remark:

What if, a keyboard similar to Microwriter is used to type Devanagari?

In a Microwriter, the hand movement is negligible, as the hand rests on the board and there is a key for every finger, the key combinations are such that no finger shifting is required except for thumb. If such text input method is applied to a script such as Devanagari, which has a large number of characters, the number of keys has to be increased. Just 5 keys (the 6th one in Microwriter is used for changing mode [17]) will not be able to accommodate 53 base letters.

For example, if one more key is added i.e.. If 6 keys are used for text entry (other than mode change key), by using maximum of 5 fingers for 6 keys, 62 combinations can be achieved which can accommodate the 53 base letters of Devanagari.
But these will also involve awkward combinations of fingers, for e.g.. (Index + ring finger), (thumb + middle + little finger) etc. which will not be easy to press and will add to the errors. Having more keys will lead to finger travel and will add hand movement.

Also, in Microwriter, the key combinations are assigned to letters based on their form (fig. 15). There is a possibility that because of large character size, this pattern might not work with Devanagari. As Devanagari has a unique phonetic structure, so defining a pattern based on the structure of script for assigning key combinations for letters might work better in this case.

4.2.5 Frog Pad

Frog Pad (Fig.16) is a one-handed chorded input keyboard, the size of which is comparable to the numeric keypad of a standard QWERTY keyboard for desktop. It is optimized by character frequency. “85% of average keystrokes in English text can be typed without chording, and chords are limited to 2 fingers (opposable thumb and one finger).” [18] [19]

The keyboard has more number of keys than a Microwriter or a Cykey Keyboard; it consists of 15 centrally located keys plus 5 keys along the lower edge. Keys can represent different characters depending on the mode the keyboard is in and any chording, if used.

Figure 16: Frog pad
Source: www.digital501.com [20]

The figure below shows the normal rest position of the five fingers of the hand on the keyboard.

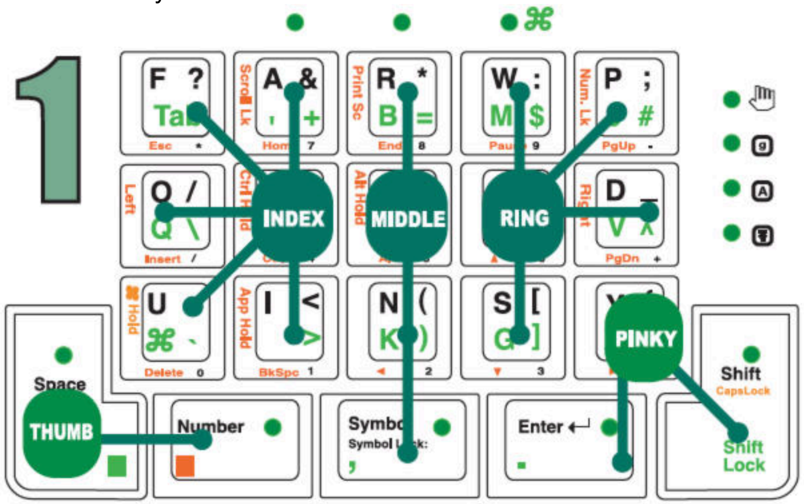


Figure 17: Normal rest position of the five fingers of the hand on the keyboard. Source: www.frogpad.com [19]

The extended lines show the finger travel that individual finger has to do while the process of typing. The keyboard is designed in a way such that the load gets distributed among the fingers. The fingers that have better pointing accuracy are allotted more space/keys to cover. Thus, amount of efforts made (reaching the key + tapping) are quite controlled in this layout.

Further, using the visuals, a demonstration is made showing how to type in a frog pad.

Figure 18: To enter primary lower case letter.
Source: www.frogpad.com [19]

The letters in black (fig. 18) are in the first layer that is by default active. The letter is entered when button is pressed and released.

In order to enter the letter in green (fig. 19), the green thumb key is pressed simultaneously with the required letter key.

Figure 19: To enter secondary lower case letters.
Source: www.frogpad.com [19]

The most important thing here is that it does not matter which key is pressed first, the letter gets enter when the pressed keys are released. This feature will reduce the error significantly as need for doing sequential operations doesn't exist.

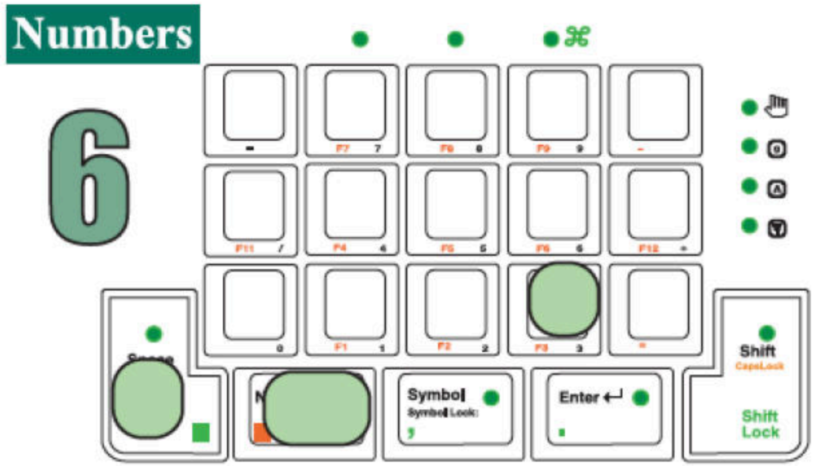


Figure 20: To enter the numbers and functions.
Source: www.frogpad.com [19]

Pressing the number key will initiate the number mode. Thus to enter the numbers, the required key is pressed. To get into the function mode for e.g.. F1, F2 and so on, Number key is pressed simultaneously with the space key.

4.3 Text input methods in Devanagari

4.3.1 InScript Keyboard

Fig.21: Layout of INSCRIPT Keyboard

InScript keyboard (Fig. 21) follows the Qwerty keyboard layout with Devanagari characters mapped on it. To deal with the challenge of large set of letters, two letters are assigned to one key. The letters mapped on the bottom-right are on the default layer, while the ones on top-left of the key are accessible on continuous pressing of SHIFT key which is a sequential operation, which means first shift key has to be pressed followed by the required key to enter the letter on top-left. [21]

The distribution of letters is such that the more frequent ones are kept on the bottom-right positions of the key in the default layer. Also, the vowels are mapped on the left side that enhances the possibility of alternative hand typing. [7]

The use of Shift key increases the number of keystrokes. As discussed earlier, it is a sequential operation here, which involves proper attention while pressing keys, else errors will occur. The letters are not distributed as per the logical structure of the Devanagari, so it involves hunting and pecking for a novice user. Also, many punctuations marks are not marked in the layout.

4.3.2 KeyLekh

Fig.22: Keylekh, keyboard for Devanagari.

Keylekh (fig. 22) is a desktop Keyboard for Devanagari, which can also be used to type English [5]. In Keylekh, separate keys are used for full consonants and vowels. User can toggle to pure consonants and matras using SHIFT key. Thus, the number of keystrokes required is less. Letter arrangement is according to the structure of Devanagari and their positions are such that it allows the more frequent consonants to fall on the index finger of the right hand [21]. Colors are used to differentiate between vowels and consonants. Thus, searching of desired letters is easy. [7]

4.4 Available input methods on mobile phones in Devanagari (Often used with one-hand)

4.4.1 Text input method in Nokia mobile phone

Fig 23: Text input method in Nokia. Source: [6]

It involves multiple tapping, here multiple letters are assigned to a single button and multiple keystrokes are used on a button to input the desired letter.
For roman scripts at most 4 letters are assigned to a button but for Devanagari, 5 to 12 letters are assigned. So, it is quite a problem for users to remember the sequence, sometimes the users miss the letter by over-pressing the key in hurry and this leads them to repeat the entire cycle. It involves a lot of attention as the letters are changing speedily. [6]

4.4.2 Devanagari text input method in Samsung mobile phones (Not available now)

Fig 24: Text input method in Samsung. Source: [6]

It involves two-level coding. In this method two layers are used to resolve the ambiguity of more than one letter assigned to single button. The first layer represents the phonetically grouped consonants and vowels, and the second represents the individual consonants in the group. The grouped consonants, vowels and matras are mapped to keys 1 to 7, key 8 and key 9 respectively. A digit is pressed to access a particular group and then another digit is pressed to input the desired letter (the digits assigned to individual letters in the group are displayed on the screen). [6]

Because of only two layers, a letter gets associated to a two-digit code. Thus, only two keystrokes are required to enter a letter and user might remember the code with the frequently used letters, which will enhance the typing speed. E.g.. ग is 13, म is 55 etc. [7]

4.4.3 **Devanagari text input method in Sony Ericsson mobile phones** (Not available now)

Fig 25: Text input method in Sony Ericsson. Source: [6]

In this method (fig. 25), phonetic mapping to Roman script is done. The Hindi alphabets are mapped to similar sounding roman alphabets. So, it is a form of transliteration. E.g. ड, ढ, द, ध are mapped to ‘d’.

This method requires too many key presses. The user has to be aware of which Hindi alphabets sound similar to which Roman alphabet. E.g. To select फ, which is mapped to ‘b’ (key 2), the user needs to tap through all the letters mapped to ‘a’ (which happen to be 11 in number, given the variety of ways in which ‘a’ is pronounced in English). [6]

4.4.4 **Disha keyboard for touch screen**

This layout (fig. 26) is based on logical order of Devanagari script. A group of five characters with phonetic similarity are accessed from a single large button (for example. क, ख, ग, घ, ङ). If the phonetic similarity is not inherent with the group (for e.g.. in case of semi-vowel or vowel groups), the alphabetical order is used for grouping. The first character of the group is generally the most frequent character in the group and that is kept at the center of the button and all other characters are distributed in clockwise manner. [7] [22]

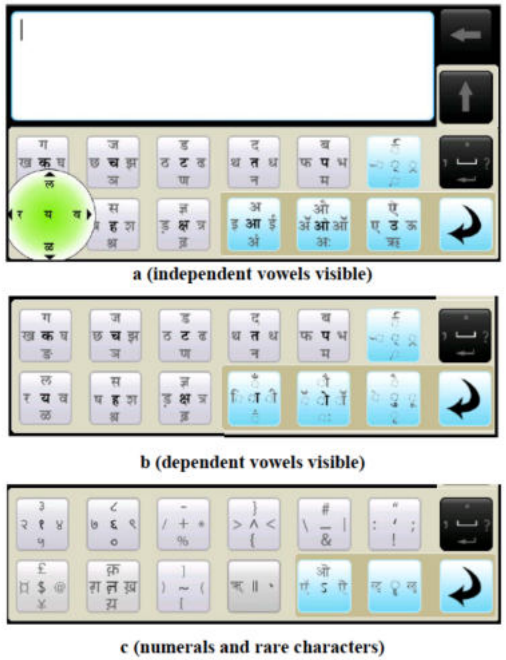


Fig 26: Disha Layout.
Source: [22]

To select a character, one has to tap the key representing the group with phonetic similarity in which the character belongs. And then the user has to slide the stylus/finger in the direction of the desired character.

As per Fitts's law, the number of keys in the layout is less, leading to larger key size. This will lead to high accuracy.

4.4.5 **Swarchakra keyboard for touch screen**

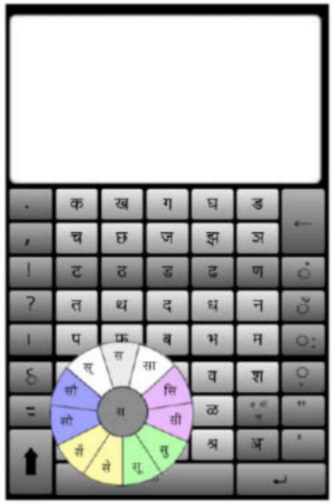
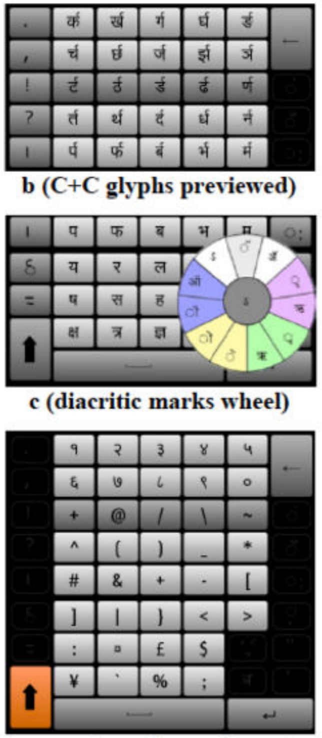


Fig 27: Swarchakra Layout.
Source: [22]

It is also a logically ordered design (Fig. 27). In this design, the default screen shows only the consonants. The layout follows the logical structure of the Devanagari script and it is arranged in a grid. When the user clicks on a consonant, a wheel with combinations of consonants plus the vowel modifiers, halant etc. pops up (fig.27) and the desired character is selected by moving the finger/stylus over it.



b (C+C glyphs previewed)

c (diacritic marks wheel)

d (numerals and rare characters)

Fig 28: Swarchakra Layout and various layers. Source: [22]

If the user wants a glyph to be selected, the key that shows the glyph is clicked and the default screen changes to preview mode showing C+C glyphs (fig. 28b) from where the desired combination can be selected.
If user wants to input a diacritic mark, then the key (fig. 28c) on the default screen has to be tapped that represents the entire group of diacritic marks and a wheel pops up from where the required symbol can be selected. Similarly for numerals and rare characters, the representative character on default screen has to be pressed (fig. 28d), the default screen changes to another screen which carries all such

characters and from where the required symbol can be entered.

The number of keys in the Swarchakra layout is a lot many as compared to Disha Layout, thus the key sizes are quite small as compared to Disha Layout. This would lead to less accuracy as compared to Disha. But, there are many merits to it. For e.g.. It follows logical structure and a complex character can be input in one slide. It was found that the logical design of Swarchakra was better for first-time usability. Where as in the study, Disha caught up with Swarchakra on speed, errors, and satisfaction during the longitudinal study. [22]
It was also established in the above-referred study [22] that if a keyboard design reflects the structure of the script, it would help users in their initial interactions and reduce the entry barriers.

5

Product brief

The following goals were identified after analysis of the challenges and studying different text input methods-

- The user should be able to use the keyboard after basic training and product demonstration/walk through.
  As discussed in section 3.4, one-handed input systems are harder to learn as compared to normal systems. Significant duration of training/product walk-through is required by the users to get used to the layout and thus, start using it. Here, the design goal is to allow the user to learn the method and use it with minimum training.

- Fast to enter text i.e. Speed of use.
  Text entry Speed must increase as quickly as possible and allow user to become an expert as early as possible. (Chances of becoming an expert must be high and quick)

- Fewer chances of errors by user

- The system should be consistent and scalable

Other checks are identified such as the design should have an optimum balance between the number of keys and key size, it should be such that the system must not compel the users to trade off between speed and accuracy. Also, the factors that enhance the chances of errors such as modality, frequent shift between the layers etc. should be avoided in the design. The factors that should be checked in the design are listed as under-

- The system must be such that the user makes brief efforts (reaching the key + tapping) to input/edit text.

- Optimum balance between finger travel and finger load, also the stronger fingers should be more frequently used.

- To minimize the cognitive load by mapping the characters on keys in a way which makes it easier for user to learn and use it.

6

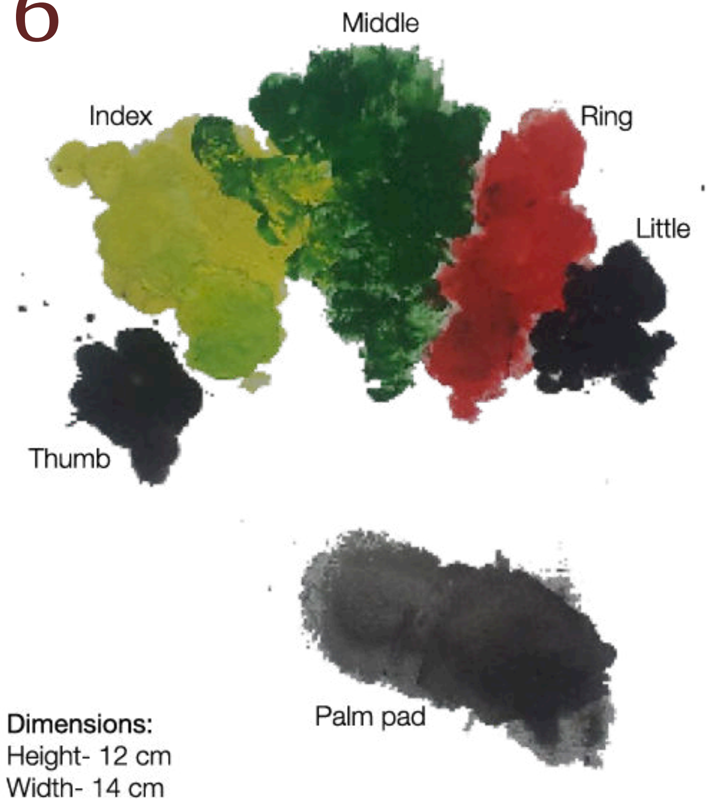


Figure 29: Impression of typing gestures of different fingers of the right hand of the student himself on a plain paper.

Concepts

The first step towards ideation was to find out what would be the rough size of the keyboard. So, as to get a rough idea of the same, different colors were applied on the fingers of the right hand and on the wrist joint of the subject (Fig. 29). Subject in this case is the student himself (body height 5’ 9”). Typing gesture was done by the fingers over the white sheet and thus, the impressions were taken. The key idea behind that was- the keyboard should allow minimum/zero hand movement.

This was the starting point of the ideation and explorations of what possibly could be the strategy of the input methods.

The initial idea was to adhere to the logical structure of the Devanagari script. As mentioned earlier, the Varnamala is learnt in India like a song, it’s sung and learnt. The structure has small phonetic groups of characters, for e.g.. (क, ख, ग, घ, ङ), (च, छ, ज, झ, ञ) and so on. If the phonetic similarity is not inherent with the group (for e.g.. in case of semi-vowel or vowel groups), the alphabetical order is used for grouping. (Similar strategy is followed with Disha Layout (fig. 29) as discussed in section 4.4.4)

Concept-1

Multiple shift-key layout

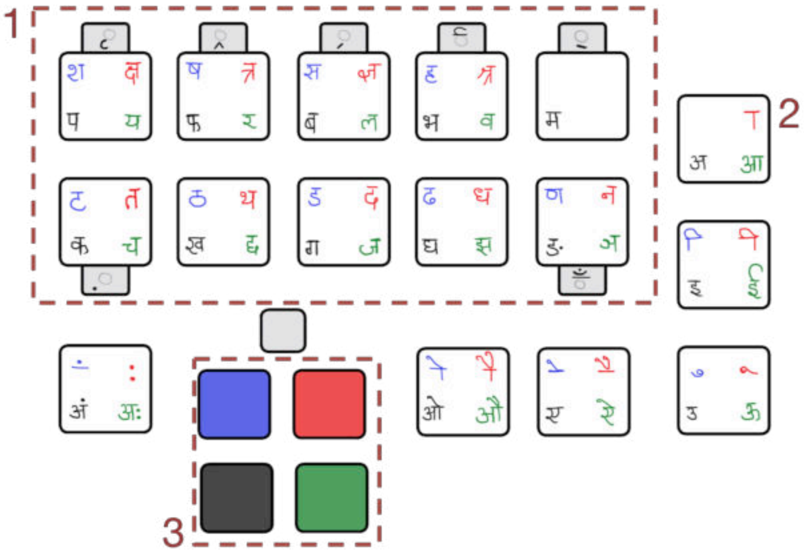


Figure 30: Initial idea of multiple shift-key layout (with various sections indicated)

In this layout, all the consonants are mapped on a set of 10 keys (Fig. 30- Section 1). This is because, group with phonetic similarity has 5 characters and there are 8 such groups of consonants (ref. to fig. 1). There is no phonetic similarity among the characters that are mapped on a key (fig. 30a) except the fact that their order of appearance in the phonetic group is the same (following the logical structure).

Figure 30a: Characters mapped on a key

Also, there are 7 special matras including halant, Chandrabindu etc. that are mapped over 7 of the 10 keys carrying these consonants (Fig. 30- Section 1).

Similarly, there are 6 keys assigned for the vowels and vowels modifiers (Fig. 30- Section 2). The lower part of the key is mapped with the vowels, where as the upper part is mapped with the vowel modifiers (Fig. 30b).

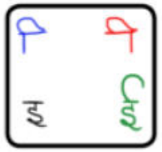


Figure 30b: Vowels and vowel modifiers are mapped on a key

Different colors are used for different phonetic group members in a row of keys (Fig. 30a). Similar color buttons in similar order are seen in Section 3 of Figure 30. These will act as ‘shift’ button to access the various planes of various groups for e.g.. (क, ख, ग, घ, ङ), (च, छ, ज, झ, ञ) etc.
This means, while RED button is pressed (in section 3 of Fig. 30), the layer containing all the characters in RED is activated, so by pressing COLORED SHIFT KEY and then pressing the key on which the desired character is mapped, the character will get printed. That means there are 5 different planes/layers in this layout, which can be accessed through the COLORED SHIFT KEY buttons.

While designing, the position of fingers was thought in a way such that the THUMB will be placed over the cluster of colored shift keys and will move among those five keys only.

After observing the users while they were typing using the paper prototype, following were the insights- the current position of the thumb was not natural, and the thumb was in constant strain.

So, following this insight, iterations were done (fig. 31).

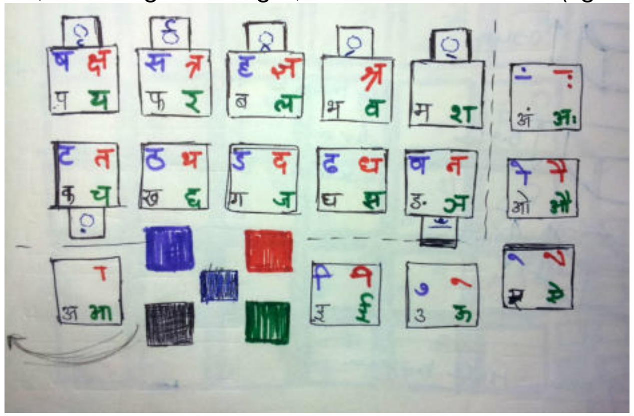


Figure 31: Iteration of Multiple Shift-Key layout

It was further iterated and in this version (fig. 32), the shift key for special matras (total 7 in number) was removed, all the special matras were put on the default plane that means for adding a special matra (or diacritical marks) to a consonant, just a press on the respective key is needed. Also, the arrangement of the keys was changed.

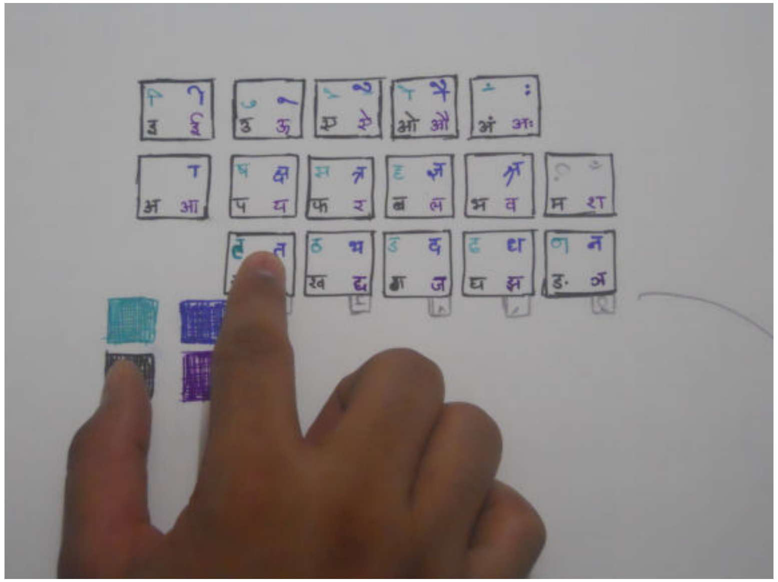


Figure 32: Final iteration of Four Shifts layout

The problem with this layout is that there is excessive load on the thumb, i.e.. With almost every character input, the colored shift key has to be pressed. Also, the pointing accuracy of the thumb is not as good as the index finger while the load is maximum on thumb, continuous movement among the shift key cluster will require a lot of concentration/attention which will in turn affect the speed and will also cause errors.

Concept-2

18 key grid-based Board

Concept-1 has a total of 20 keys. The objective for the second ideation was to reduce the number of keys even further such that the size of the keys could be increased leading to better results (following Fitts's law), and also to reduce the number of shifts as that was increasing the finger load.

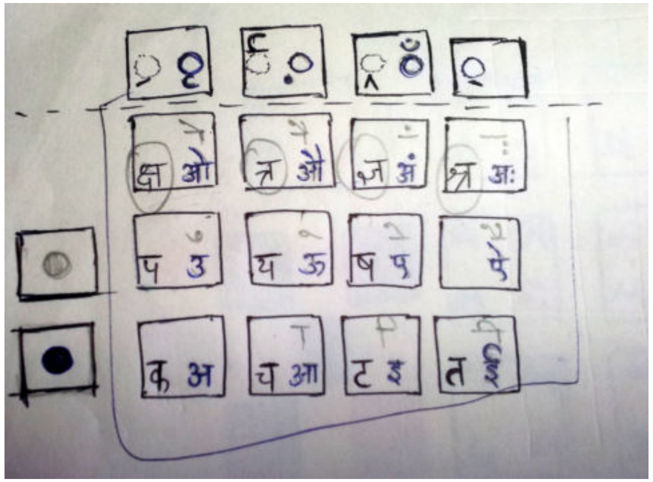


Figure 33: 18 key grid-based Board

18 keys were used to accommodate all the characters, used two level shifts. The thumb will be positioned over the leftmost buttons (two in number), lower one is blue and the upper one is grey (fig. 33).

In black ink on each key, the first letter (as per order in the structure) of the groups with phonetic similarity is mapped. Also, the layer in the black ink is the default layer. In blue ink, the vowels are mapped. And in grey color, the vowel modifiers can be seen.
The following set of visuals will give a demonstration on how this layout can be used-

As soon as the first letter of the group is pressed (fig. 34), for example.. if 'क' is pressed, the row of the keys above this key gets activated and the set of four keys highlighted in yellow will represent other members of the group, i.e.. 'ख, ग, घ, ङ' respectively. Note- the row is activated while key with 'क' mapped on it is pressed. If it is pressed and released 'क' gets printed.

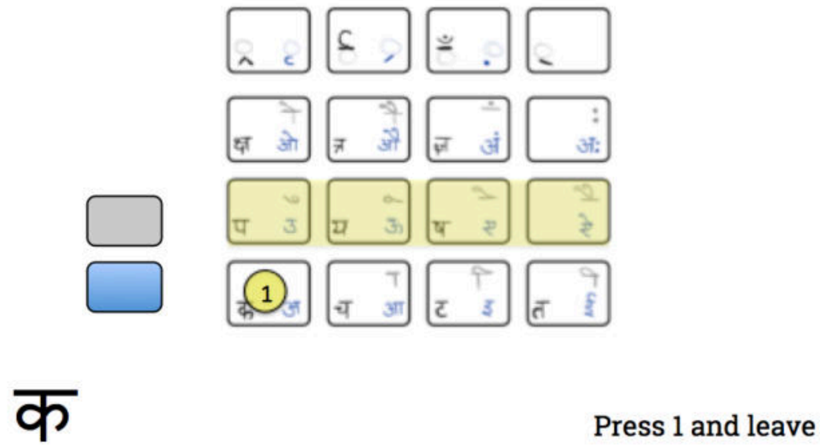


Figure 34: Demonstration of 18 key grid-based Board

While key with 'क' mapped on it is pressed, if another key in the activated row is pressed it will print the respective character (as shown in the following figure 35).

This layout has two layers, which will be activated using the two shift keys, while the every key on the lower two rows that carries the first character of the group acts as a shift key to access the other characters of the group. In this type of layout, for ease of use, it is required of the user that he has the mental model of the structure of Devanagari. Also, the number of keystrokes would be more as just 14 characters in black ink will be printed with just one key press while all other will require two key presses at least and such characters include a lot of frequently used characters.

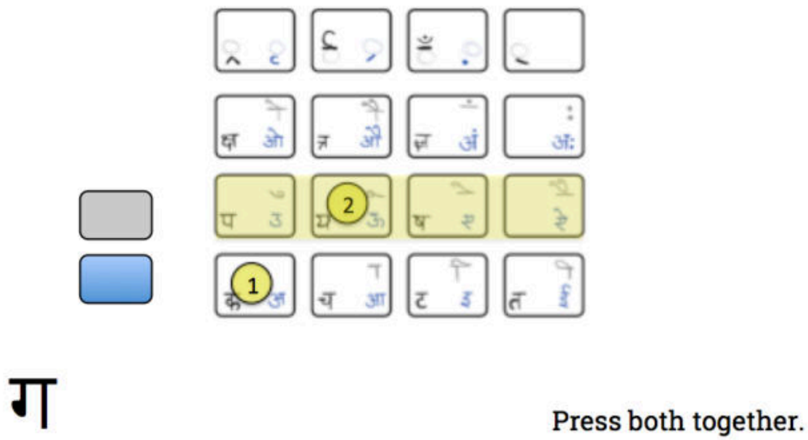


Figure 35: Demonstration of 18 key grid-based Board

Figure 36 shows the use of shift key to enter a vowel modifier.

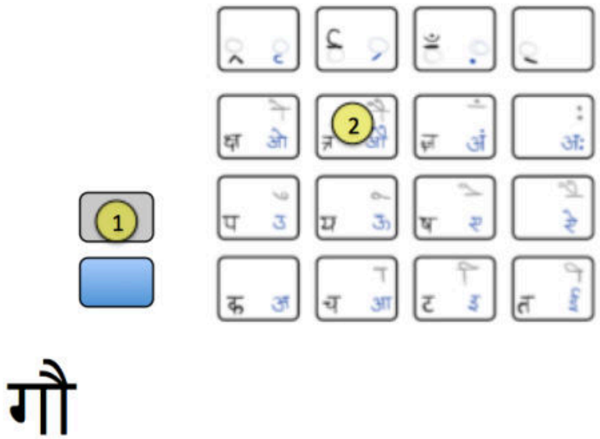


Figure 36: Demonstration of 18 key grid-based Board

Concept-3

Two-step grid-based Board

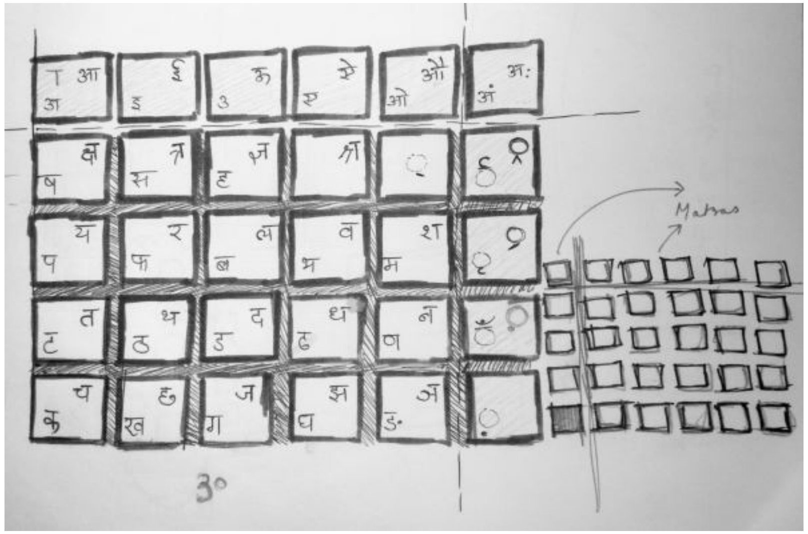


Figure 37: Two-step grid-based Board

In this layout (Fig. 37), the main idea was to eliminate 'shift' i.e.. The idea was to substitute the special button that is used to access various layers of the layout with some other gesture.

In here, two characters are mapped on a single key; a different row was allotted to the vowels and a specific section for the special matras/ diacritical marks.

The hardware keys used in the keyboard are special buttons (fig. 38), popularly called as two-step push-to-on buttons, which means these keys can accommodate two kind of key presses-i.e.. Half key press and full key press. Figure 38 can better explain the mechanism.

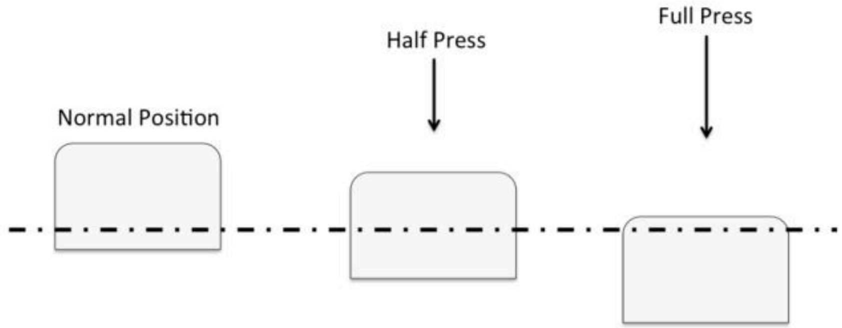


Figure 38: Two-step push-to-on button

The concept is based on a grid (5 rows and 6 columns).

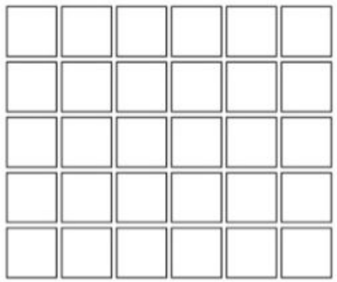


Figure 39: Grid based layout

It is calculated that for the grid-based layout, a total of 30 keys can accommodate the main characters of Devanagari script including the vowel modifiers, popular conjuncts and diacritic marks.

As discussed earlier, two letters are mapped on one key. The following figure (Fig. 40) demonstrates the character distribution on the keys-

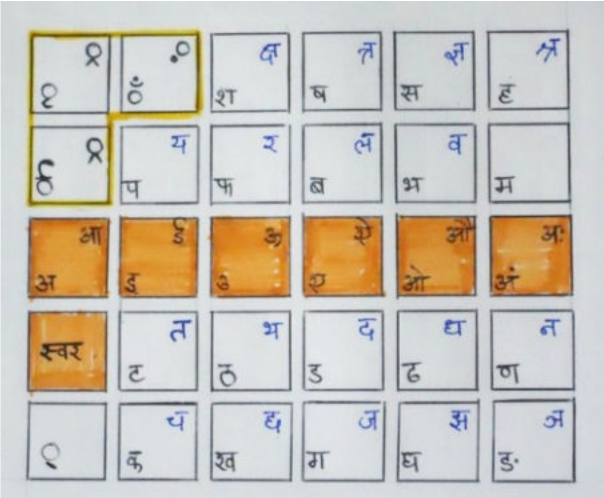


Figure 40: Two-step grid based layout

The character distribution on keys is based on the logical structure of Devanagari. In every row, the characters in similar color ink belong to the same phonetic group (the characters having phonetic similarity).
The vowels are allotted central row. Also, the diacritical marks are on the cluster of 3 buttons on the top left (Leftmost section in Fig. 40).

For getting the characters that are mapped on keys in black ink, the user has to go for half-press of push-to-on buttons. Where as for getting the characters that are mapped on keys in blue ink, the user has to go for full-press of push-to-on buttons (fig. 41).

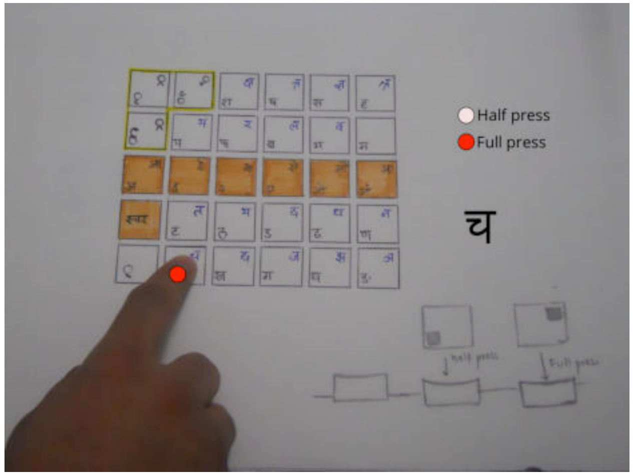


Figure 41: Demonstration of Two-step grid based layout

If the user wants to input matra/vowel modifiers on the consonants (fig. 42), the user has to press the vowel key corresponding to the required vowel modifier/matra, i.e.. The vowel modifiers are by default activated on the row of the vowels. It is demonstrated in the following figure.

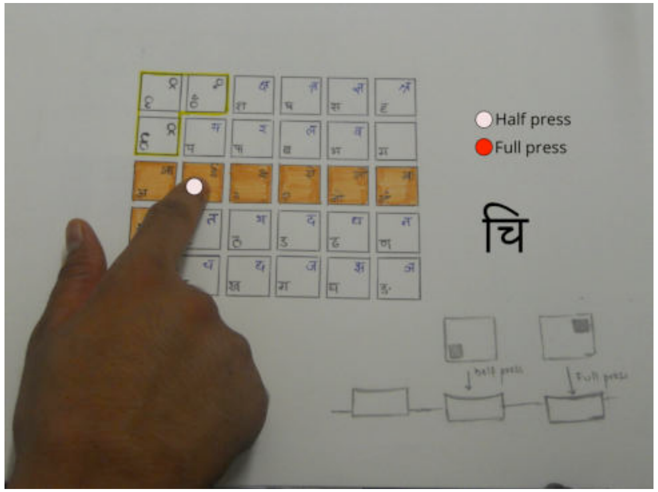


Figure 42: Putting the vowel modifiers

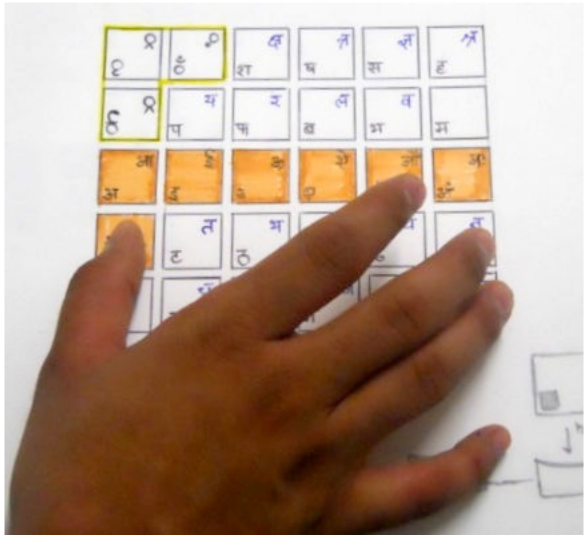


Figure 43: Activating the vowels by pressing the swara key.

For printing a vowel, layout has a shift key (mentioned as swara in the layout), which is used to activate the plane of vowels (fig. 43). The user has to first keep pressing the shift key followed by pressing the required vowel mapped on the key in the row of vowels.

Analysis of the concept-

The strength of Thumb and Index finger is higher than middle finger, followed by ring finger and little finger. Also, the pointing accuracy of Index finger is highest followed by Middle finger, thumb, ring finger and little finger.

Figure 43a: Right hand

'Alpaprana' characters are more frequently occurring than 'Mahaprana' characters (shown in figure 3). In the layout (figure 40), frequently occurring 'Alpaprana' characters lie on the same key, for e.g.. [क च]. Similarly, 'Mahaprana' characters lie together, for e.g.. [ख छ]. This will lead to a situation where one key will be very frequently used and the neighboring key will not be used that often, and this will happen for every alternate keys. Also, the frequently occurring characters are mapped in a way that will require excessive efforts from fingers with lesser strength and pointing accuracy.

Based on the above discussion, a different arrangement of characters on the keys (Fig. 43b) may give better results.

Figure 43b: Different way of character mapping

After observing the users while they were typing using the paper prototype, following was the insight- the current arrangement of vowels and vowel modifiers in the central row will involve fingers to stretch in and stretch out quite rigorously, which is an awkward gesture.

Concept-4

Microwriter based layout

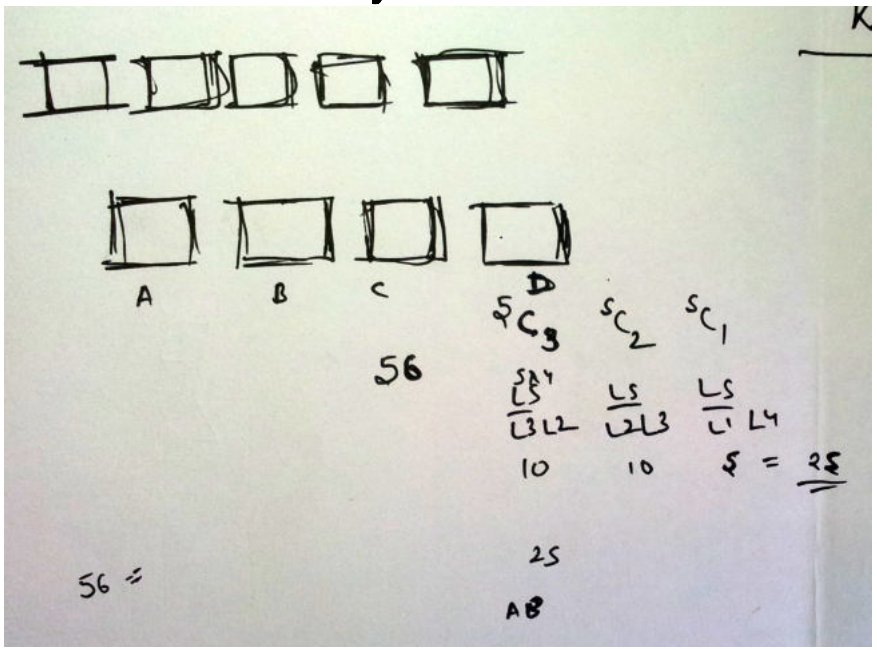


Figure 44: Calculating the minimum number of keys required for a chorded keyboard like a Microwriter.

The idea was to conceptualize a chorded keyboard such as Microwriter, Cykey etc. for Devanagari. The initial challenge was to calculate the minimum number of keys that can be used to accommodate the base letters of Devanagari. Various combinations were tried for e.g.. Using 7 keys, 8 keys, 9 keys and so on. The final iteration showed that after avoiding awkward finger combinations such as the entire set of base letters could be assigned a key combination using various combinations of 8 keys. The awkward finger combinations (which were not easy to press) for e.g.. (Index + ring finger), (thumb + middle + little finger) etc. were avoided while calculating for the number of keys.

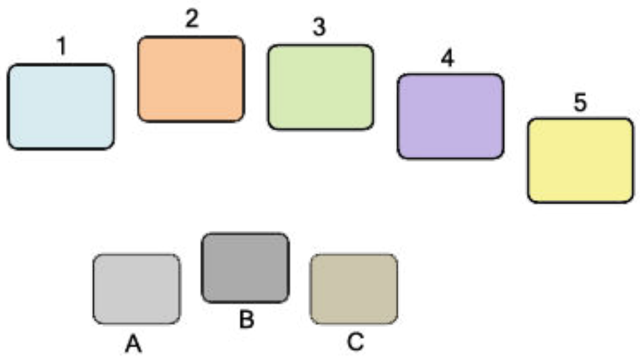


Figure 45: Microwriter based layout for Right hand

In the above 8 keys layout (fig. 45), the thumb will be placed over keys A, B and C and the remaining four fingers of the hand will be placed over the keys 1, 2, 3, 4 and 5.

A, B and C will be called as Thumb keys.
And 1, 2, 3, 4 and 5 will be called Finger keys for the sake of convenience in the description.

For text entry, simultaneous presses of thumb keys and finger keys will be required, or just the finger keys. The order of the key-press will not matter, as the text will be entered as soon as all keys is released. Various combinations can be seen in figure 46.

Key A + combinations of finger keys will give one set of consonants.
Key B + combinations of finger keys will give one another set of consonants.

Key A + Key B + combinations of finger keys will give Vowels.

Combination of finger keys will give vowel modifiers; and so on.

Also, unlike Microwriter, the pattern for assigning the key combinations to different letters is based on the phonetic structure of Devanagari script (fig. 46).

Microwriter has an advantage over this, it had 6 keys, and while typing the position of fingers and thumb was fixed over a particular key, where as in this (fig. 45), a lot of finger movement has to take place.

There a large number of characters in Devanagari, even though it follows obvious pattern based on Hindi varnamala, even then remembering such combinations would put a lot of cognitive load on the novice/ beginner.

Figure 46: Key combinations for Microwriter based layout for Right hand

Concept-5

Melodica based layout- ‘Melo Keyboard’

Inspired from a Melodica, the idea involved multiple iterations that included figuring out the finger combinations for text entry, finding optimum number of keys required for accommodating all the letters.

Considering the strength of the individual fingers and their accuracy in pointing, it was decided that combinations involving all five fingers of the hand will not be considered unlike concept-4, further using four fingers to do text input was also avoided.

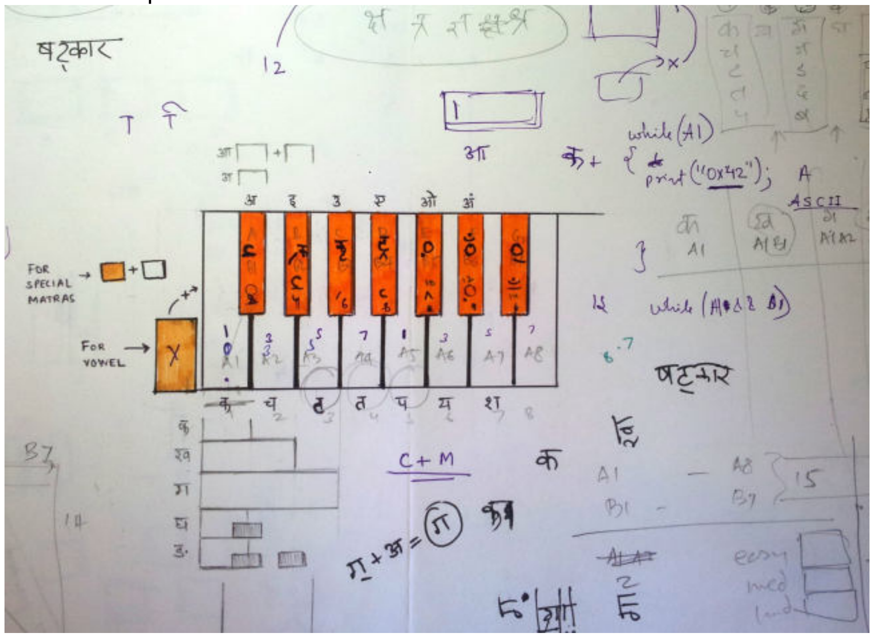


Figure 47: Figuring out the number of keys required for concept-5

Using maximum of 3-finger combination for text entry, it is calculated that by using 15 keys in the layout, all the base characters of the

Devanagari script could be accommodated (fig. 47).

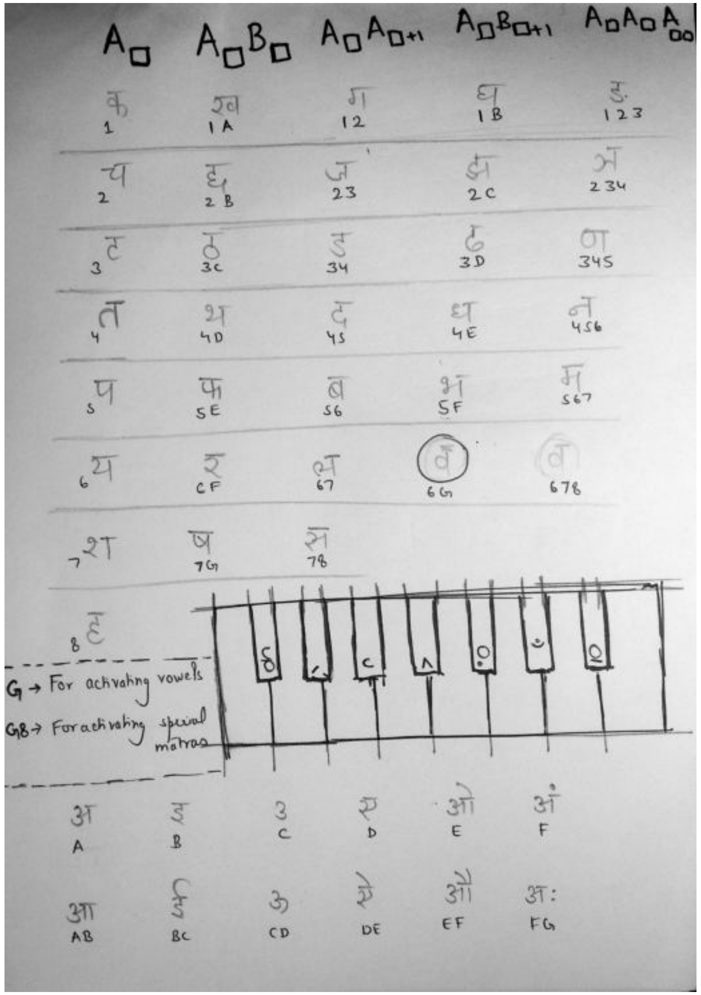


Figure 48: Figuring out key combinations for concept-5

Special attention was given such that while doing text input, the fingers should move as a package i.e.. They should not be stretched for reaching out to keys while pressing key combinations for inputting a single character. (Fig. 48)

Layout and arrangement of the keys

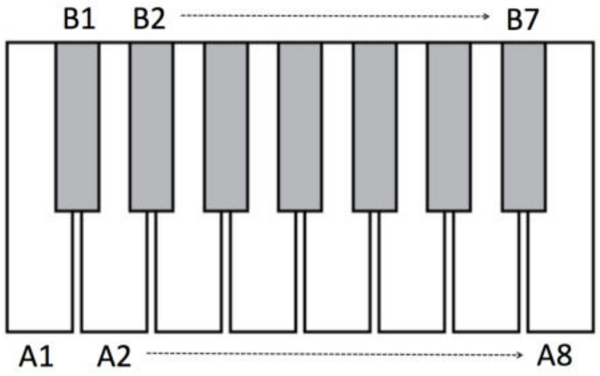


Figure 49

This is inspired by the design of Melodica, but unlike a melodica this has continuous set of 7 black keys and 8 white keys (Fig. 49).
The current dimensions of the concept are based on the dimension of the melodica, which was the trigger for the inspiration (Fig. 50).

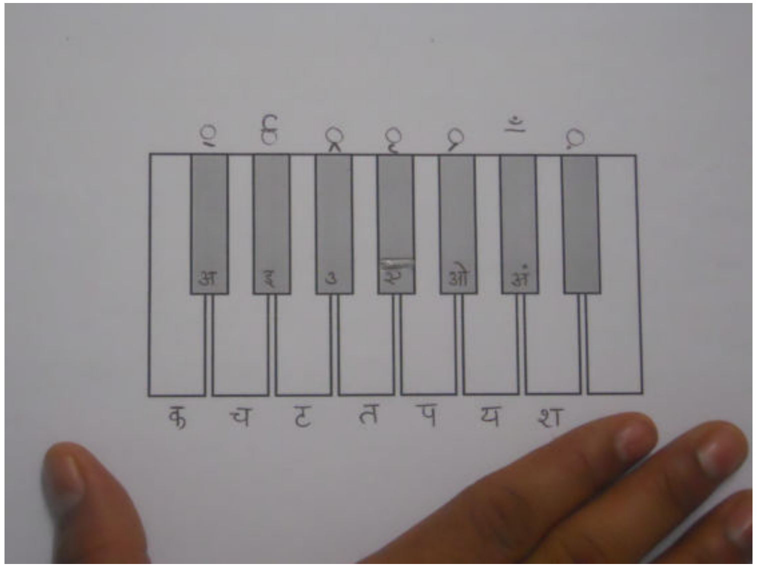


Figure 50: Demonstrating the rough dimensions of the layout.

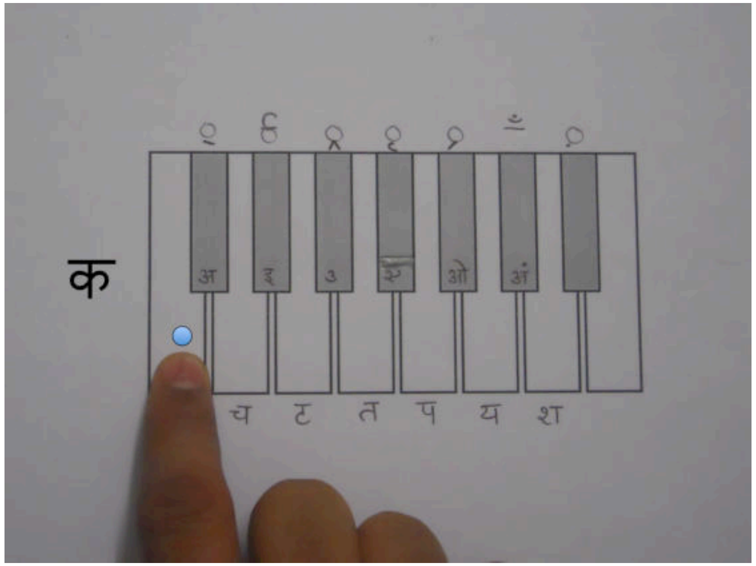


Figure 51: Keyboard Demo

The concept constitutes of **3 layers**-

The default layer (PLANE 1) is of the consonants and the matras, the reason is that they are more frequently used as compared to Vowels and diacritic marks.

So, if one presses any of the A1, A2, … , A8, a set of consonants can be achieved.
For example, pressing A1 gives क (Fig. 51)

While using combinations of
A1 (pressed) followed by B1 gives ख (Fig. 52),
A1 (pressed) followed by A2 gives ग,
A1 (pressed) followed by B2 gives घ,
A1 (pressed) followed by A2 and then A3 gives ङ

Similarly, pressing A2 gives च.
And using A2 with other key combinations gives other letters similarly.

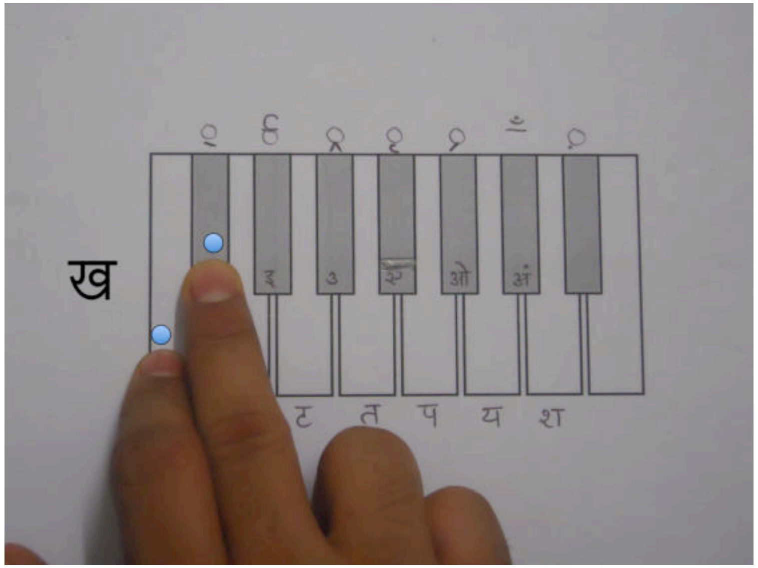


Figure 52: A1 (pressed) followed by B1 gives ख.

Referring to figure 3, classification of a phonetic group in Devanagari script can be broadly seen under the aspiration category, namely- alpaprana, and mahaprana. In the frequency-based list of letters, it is found that the alpaprana are more frequent than mahaprana.
So, in this design- the alpaprana are kept on white key or combination of white keys, i.e..
A1, A2, A3 and so on give क, च, ट.
(A1+A2), (A2+A3), (A3+A4) and so on gives ग, ज, ड..
(A1+ A2+ A3), (A2+ A3+ A4), (A3+ A4+ A5) and so on gives ङ, ञ, ण..

While,
(A1+B2), (A2+B2) and so on gives ख, छ.
(A1+B3), (A2+B3) and so on gives घ, झ..

In the default plane, for getting matra (vowel modifiers)
Pressing B1 will not give any matra.
As B1 represent the matra associated with 'अ' that does not have any representation symbol as such.
Pressing B1 followed by B2 (B1 + B2) gives ' ा '

Similarly pressing B2 gives ' ि '
Pressing B2 followed by B3 (B2 + B3) gives ' ी '

As shown in the figure, similarly B3 is mapped with ' ु '
Pressing B3 + B4 gives ' ू ' and so on.

The other layer (PLANE 2) is the layer that contains all the vowels. To change the mode B7 is pressed followed by A8. This changes it to vowel input mode (Fig. 53).

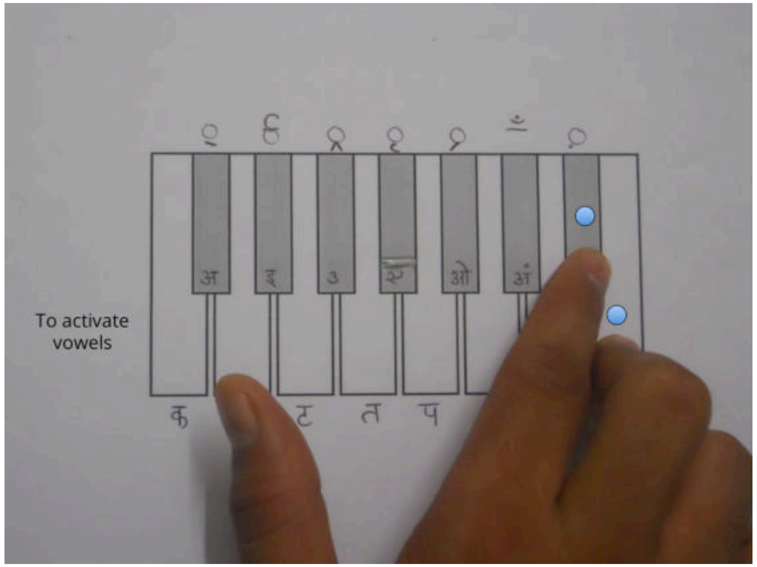


Fig. 53: Change of mode- Vowels

And now the vowels are activated for just one time press,
Pressing B1 gives ‘ अ ’
Pressing B1 followed by B2 i.e.. (B1 + B2) gives ‘ आ ’

Similarly pressing B2 gives ‘ इ ’
Pressing B2 followed by B3 (B2 + B3) gives ‘ ई ’

It follows the same pattern as it followed for the matras (vowel modifiers), but in this case, instead of vowel modifiers, vowels will be printed.

The last layer (PLANE 3) is the layer, which contains all the diacritic marks are there. Hereby it will be called special matras. To change the mode to diacritic mark input mode, B7 is pressed. (Fig. 54)

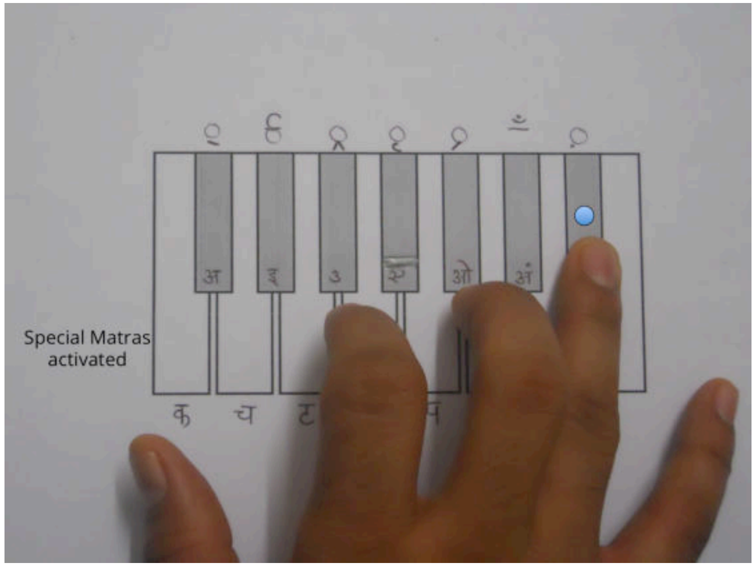


Fig. 54: Change of mode- Special Matras

After B7 is pressed, the diacritic marks (special matras) are activated for just one time press.

For instance,
Pressing B4 gives ‘ ृ ’
Pressing B6 gives ‘ ँ ’ ; etc..

7

Final Concept and prototype

7.1 Final Concept

The final prototyped concept is concept-5, ‘Melo keyboard’. The following is the layout of the keyboard-

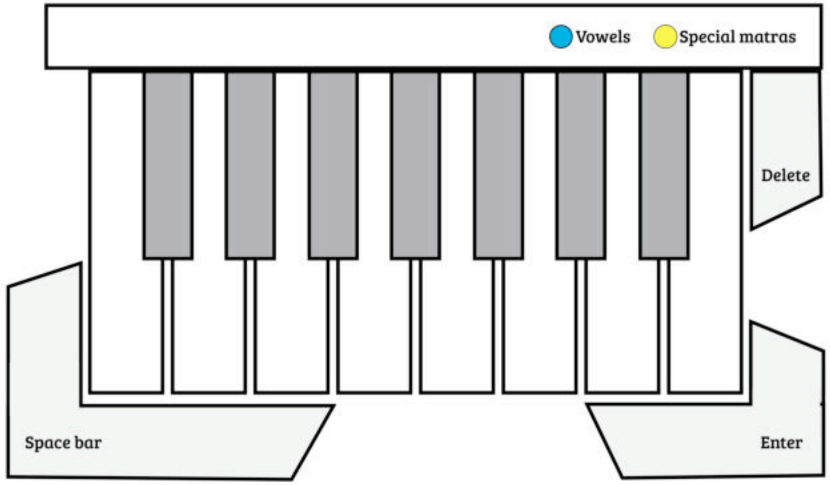


Fig. 54: Layout of the Melo Keyboard

The following visuals (figure 55) demonstrate the key combinations used to input characters in the Melo Keyboard (also, discussed under Concept-5 in section 6)

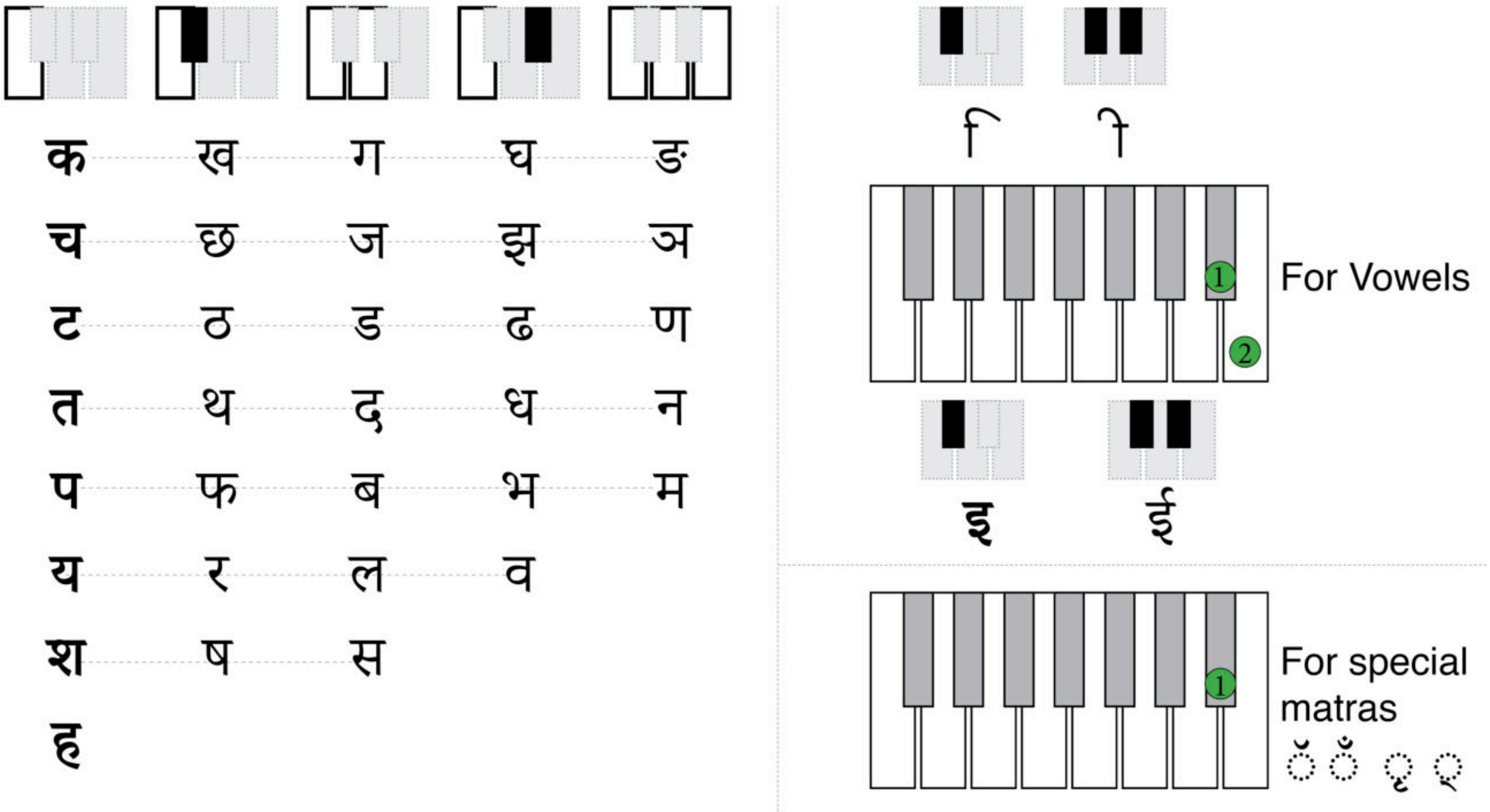


Fig. 55: Key combinations used to input letters in Melo Keyboard

7.2 **Prototype**

A piano keyboard was dismantled and the keys were rearranged according to the layout of the concept (figure 54).

Fig. 56: Rearrangement of the keys of a Piano Keyboard

Attempts were made to hack the circuit of the piano, but due to some feasibility issues, all the required functions were not being achieved. As an alternative, push-to-on button were used.

Fig. 56: Push-to on buttons

Push-to-on buttons (figure 56) were arranged under the rearranged keys.

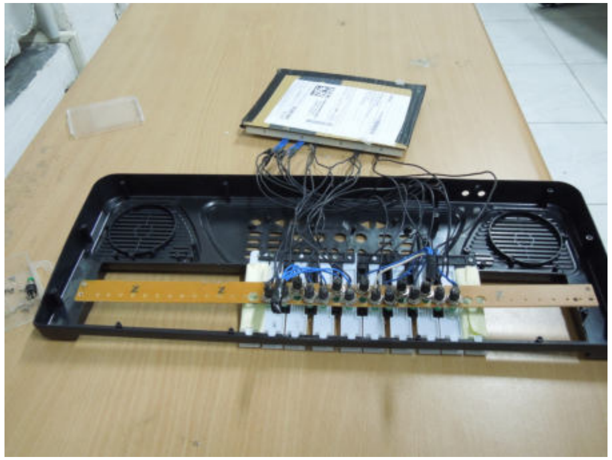


Fig. 57: Arrangement of Push-to-on buttons under the keys

The input from the push-to-on buttons was directed to Arduino Due Board, and output from the Arduino board was input to the computer.

Fig. 58: The keyboard set-up

The Arduino Due Board was programmed (figure 59) such that the various key combinations on push-to-on buttons gave Roman letters as output on the computer, which was mapped against their Devanagari Unicode Equivalents-which were displayed on the Computer screen (figure 60 & 61).

```
BASECODE1_LAST_BACKUP
{
   delay(10);
   if(digitalRead(1)==LOW && a1==0)
   {
      a1=1;

      for (z=0;z<1000000;z++)  // 1000,000 = 5sec
      {
        if(digitalRead(1)==HIGH)
          break;
        if (digitalRead(2)==LOW)
          {a2=1;break;}
        if  (digitalRead(4)==LOW && digitalRead(1)==LOW)
          {a4=1;break;}
        if(digitalRead(3)==LOW && digitalRead(1)==LOW)
          if(digitalRead(5)==LOW)
            {a5=1;break;}
          else
            {a3=1;}
      }

  }
}

if (a1==1 || a2==1 || a3==1 || a4==1 || a5==1)
{
    if (a1==1 && a3!=1 && a5!=1 && a2!=1 & a4!=1)
    {Keyboard.print('k');
    while(digitalRead(1)==LOW);
    a1=0;
    reset();}

    if (a2==1)
    {Keyboard.print('K');
    while(digitalRead(1)==LOW || digitalRead(2)==LOW);
    reset();}

    if (a3==1 && a5!=1)
    {Keyboard.print('i');
    while(digitalRead(1)==LOW || digitalRead(3)==LOW);
    reset();}
```

Fig. 59: Program for the Arduino Board

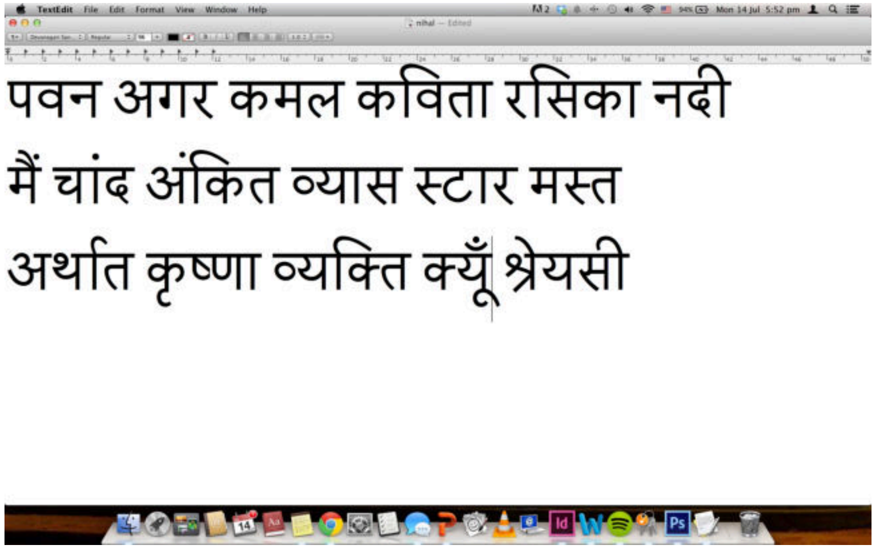


Fig. 60: Words typed on the computer using the Melo Keyboard

For facilitating the process of learning for a first-time user, a panel (figure 55) was placed over the Keyboard (figure 61) such that the user could take reference from the panel while typing the letters.

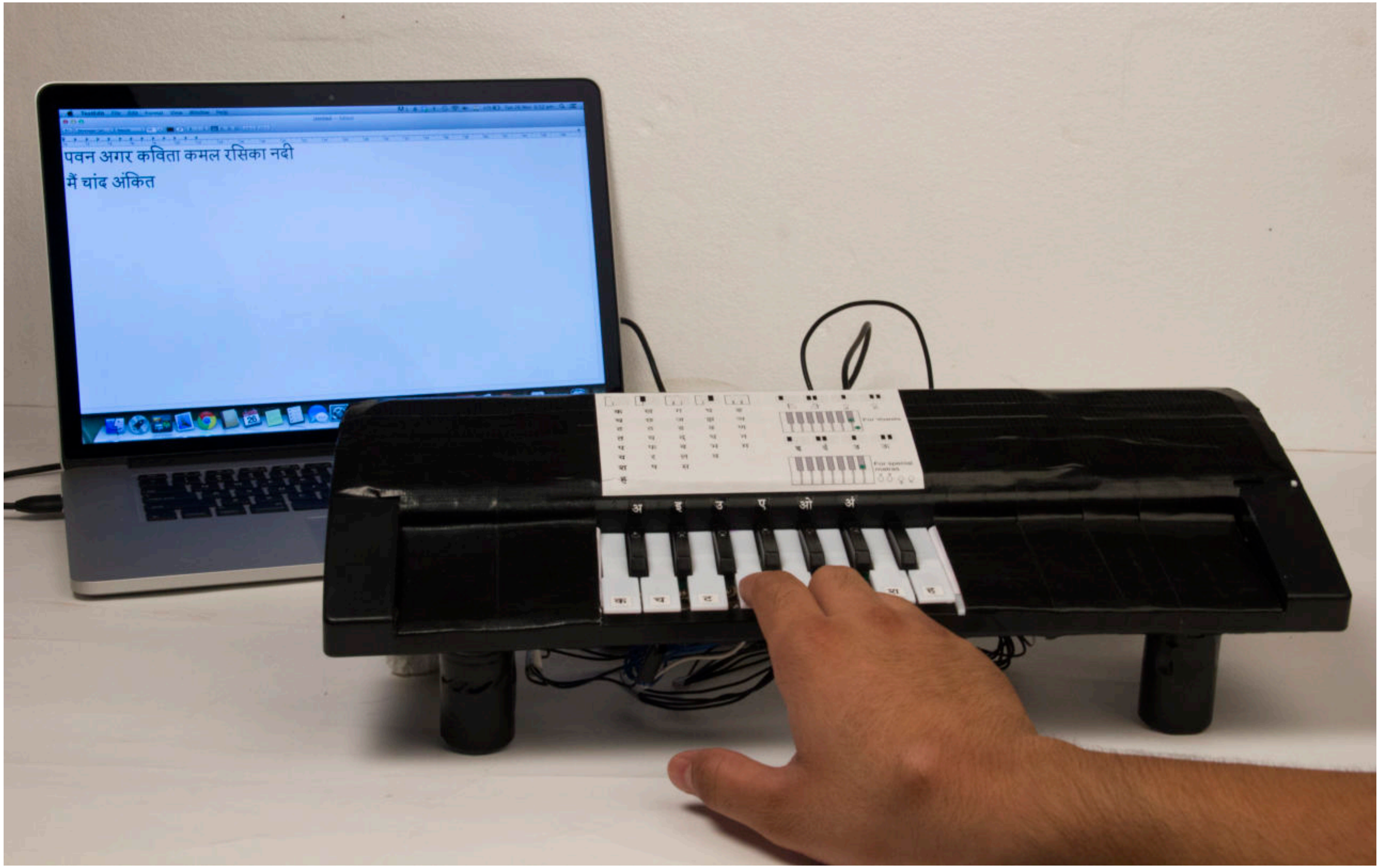


Figure 61: Final prototype of Melo Keyboard along with the panel

7.2.1 The dimensions of the keys in the prototype-

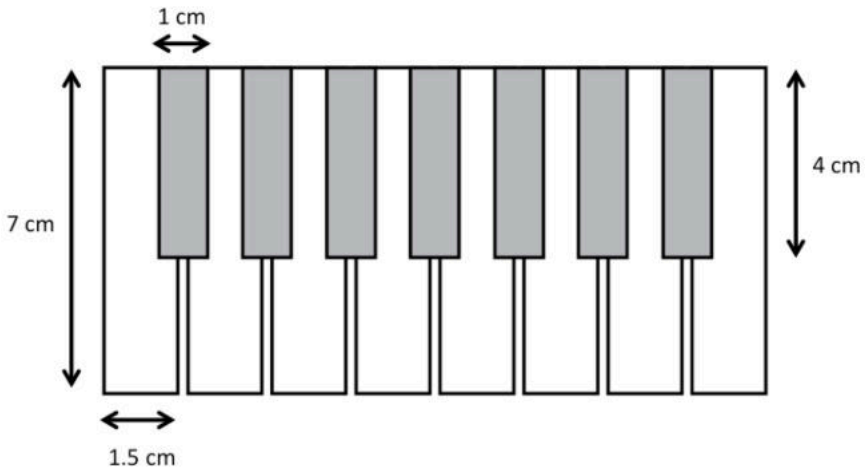


Figure 61a: Dimensions of the keys in the prototype

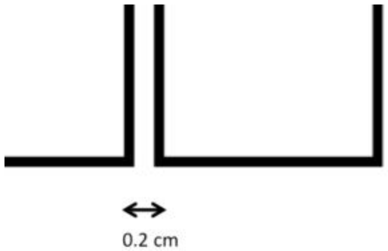


Figure 61b: Dimensions of the key spacing

7.2.2 Limitations of the prototype-

The use of push-to-on buttons made the keys harder to press. This required the users to take extra measures to avoid errors while typing. To accommodate the circuit and the buttons in the case, the Keyboard was raised to height of 3 cm from ground using columns that made it harder for the user to type as there was no base to place/rest their hands while they were typing.

The above-discussed limitations could potentially lead to increased number of errors and lower typing speed.

8

Concept evaluation

The final concept ‘Melo Keyboard’ was tested and evaluated with the required users. The user criteria for user recruitment is discussed as under-

8.1 User criteria

The following was the criteria for user-recruitment-

- Mother tongue: Hindi
- Studied Hindi as a subject till 10th standard.
  -Must know to read and write in Hindi.
  -Must know Hindi Varnamala.
  If not, provide assistance to recall.
- Education: Class 12th and above.
- Age: 18 to 25 years.
- Must not have used any text input system in Devanagari.

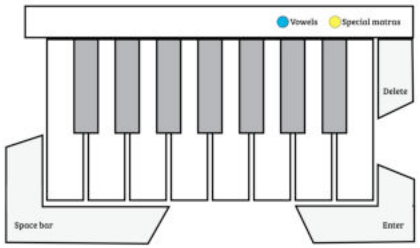
8.2 Different evaluations conducted

Two types of evaluations were primarily conducted. The details are discussed below-

1. **Theoretical evaluation**
   Theoretical evaluation was conducted on Melo-Keyboard, Inscript keyboard for One-hand use and Half-Qwerty Keyboard (one-hand input in Devanagari).

2. **Empirical evaluation**
   a. **Short term testing** (Melo V/s InScript)
   b. **Long term testing** (Melo V/s InScript)

8.2.1 **Theoretical evaluation**

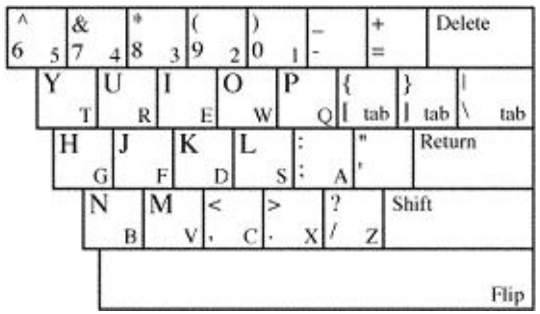


Melo Keyboard

V/s

InScript (used with One-hand)

V/s



Half Qwerty for Right hand (used for input in Devanagari)

8.2.1a **Goal of Theoretical Evaluation-**

The goal of Theoretical evaluation was to measure the overall effort in Melo Keyboard as compared to the InScript keyboard (when used with one hand) and Half-Qwerty keyboard (when used with one hand for input in Devanagari). The Overall effort comprises of the following parameters-

a) Number of Key presses
Measuring the number of key presses for the given text.

b) Finger travel
Measuring the distance travelled by fingers over the keyboard for doing key presses for entering the given text.

c) Finger load
Measuring the effort done by the individual fingers for entering the given text.

Comparison of-Melo with InScript and Half Qwerty-

There are no specially designed keyboards available for one-handed text input in Devanagari. So, Melo-Keyboard is compared with InScript Keyboard (when used with one-hand) and with Half-qwerty Keyboard (when used to input in Devanagari).

8.2.1b **Methodology for Theoretical Evaluation-**

The Keyboards were tested for **right hand** only. The Method for calculating various parameters (section 8.2.1a) in measuring the Overall effort is discussed as under-

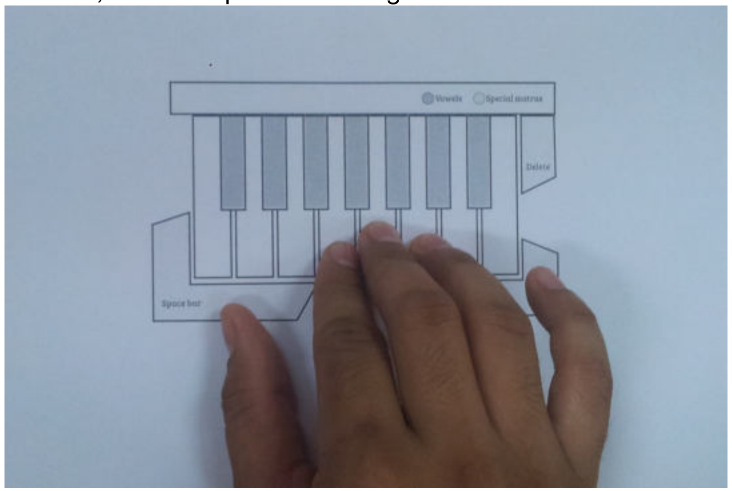
The evaluation was done for three sets of words that had different level of complexities-

Set 1: Easy
कमल पवन भारत रसिका और मित्र हैं

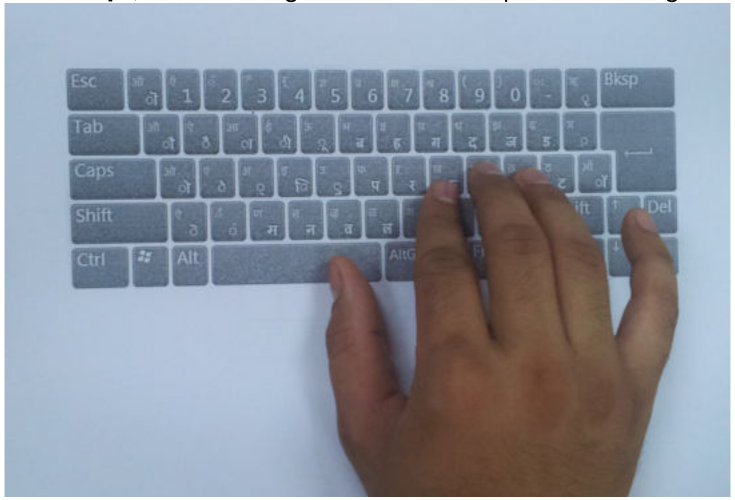
Set 2: Medium
गणपति स्टार प्रयास कूँय स्थान

Set 3: Difficult
शर्मा क्षत्रिय आर्य त्रिशंकु कृष्णा

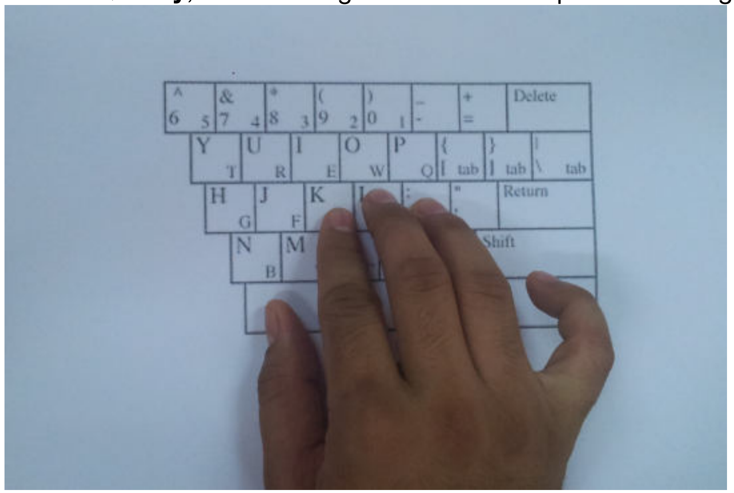
The fingers of the right hand were set to defined positions before starting to type individual words in the above-mentioned sets.
In **Melo**, the initial position of fingers is illustrated below-



In **InScript**, the following will be the initial position of fingers-



In **Half-Qwerty**, the following will be the initial position of fingers-



Calculating finger-travel:
Taking these positions as the starting points, finger travel was calculated. For e.g..

To **type 'कमल'** in **Melo** Keyboard-

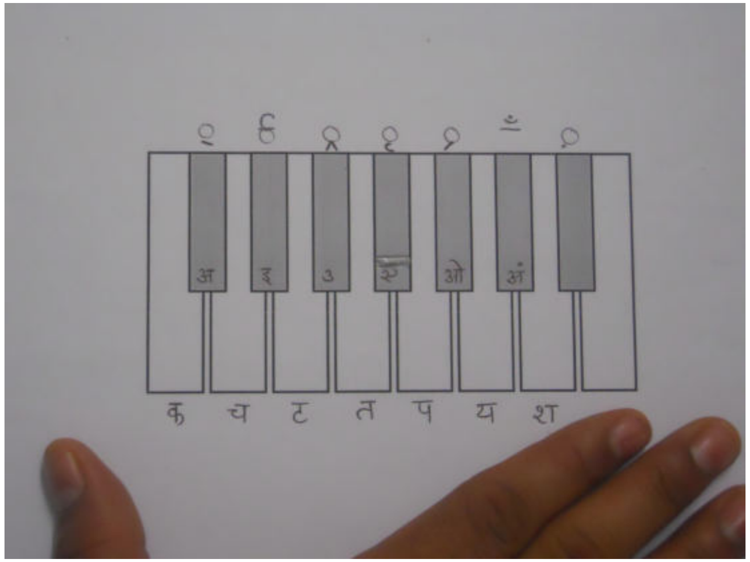


Figure 62: Melo-Keyboard

Initial position of index finger is on 4th white key, middle finger is on 5th white key and ring finger is on 6th key (Section 8.2.1b).

To type 'क', index finger moves from 4th key to 1st key; and is displaced by 4.5 cm.
To type 'म', Index finger has to move from 1st to 5th key, also middle finger and ring finger has to move to 6th and 7th key respectively, but the fingers are moving together i.e.. There is negligible spread between fingers. In this case, the displacement is measured (as average for all 3 fingers) from their position while typing 'क', and not individually. Here, the fingers are displaced by 4.5 cm.
To type 'ल', middle and ring finger can type it without any finger movement.

So, the total finger travel to type 'कमल' is 9 cm.

Calculating Number of key-presses and finger load-

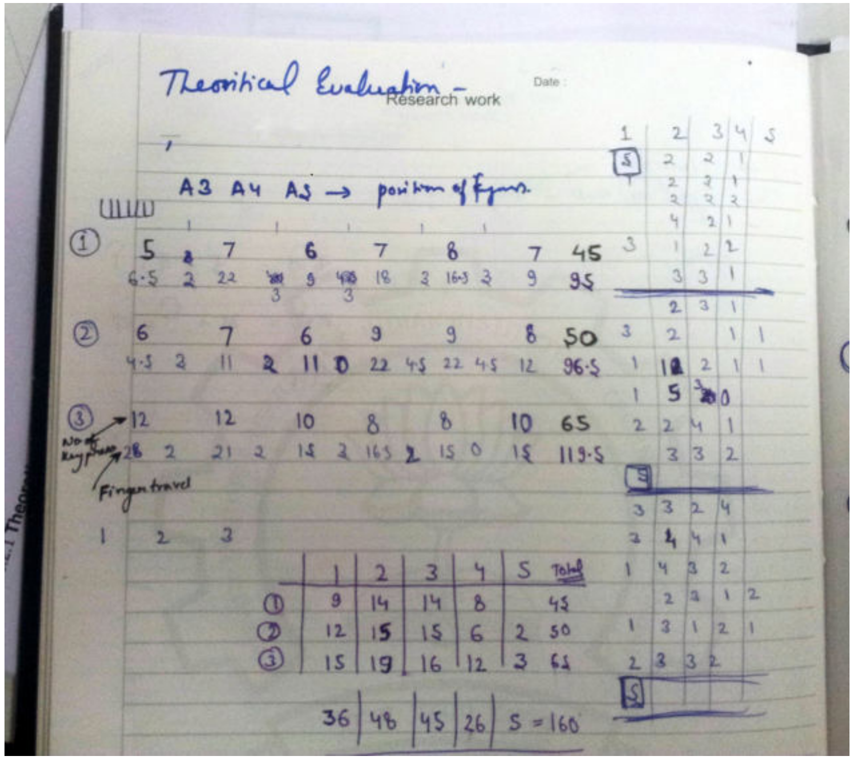


Figure 63: Calculating the no. of key presses, finger travel and finger load for Melo Keyboard for the 3 sets of words

While typing a word, the number of key presses required to type that word was noted. For example,
To type 'कमल' in **Melo** Keyboard, 6 Key presses were required.
To type 'कमल' in **InScript** Keyboard, 3 Key presses were required.

So as to calculate finger load, details were recorded about the finger that was used to perform the key presses and its frequency of use. The data about finger load will vary if the initial position of the fingers is changed, that is why the initial position of fingers (section 8.2.1b) was fixed for different keyboards for the evaluation.
For example,
To type 'कमल' in **Melo** Keyboard, the frequency of use of individual fingers is listed below-
Index finger- 3
Middle finger- 2
Ring finger- 1

To type 'कमल' in **InScript** Keyboard, the frequency of use of individual fingers is listed below-
Index finger- 2
Middle finger- 1

8.2.1b **Results of Theoretical evaluation**

1. Key-press and Finger travel

OVERALL

	MELO	INSCRIPT	HALF-QWERTY
KEY PRESS	124	85	121
FINGER TRAVEL	257	402	258

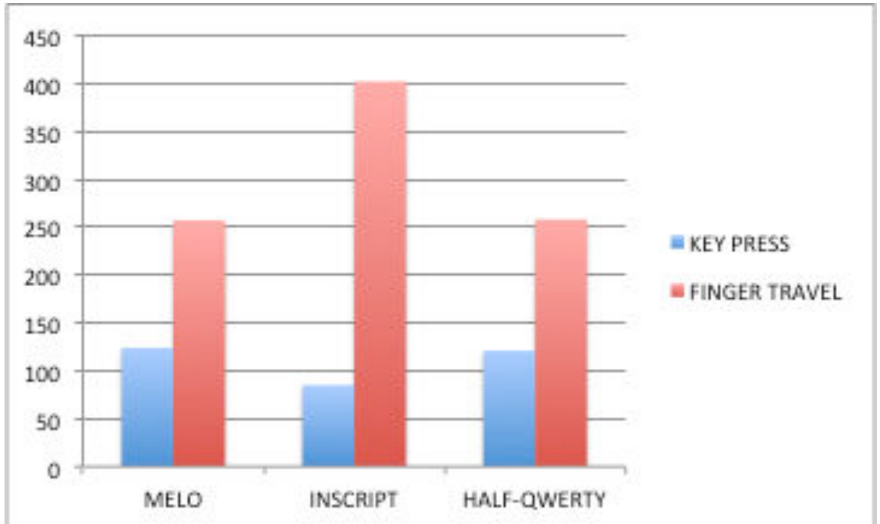


Table 1: Comparison for no. of key presses and finger travel for all three set of words.

The number of key-presses for InScript (when used for one-handed input) was the least but, it involved a lot of finger travel as compared to Melo and Half-qwerty. Also, because it was used with one-hand it required difficult hand gestures to type.

For e.g.. To type ‘ख’ in InScript, shift key is pressed using thumb and index finger is stretched to press key with ‘ख’ (shift + k), which is an awkward and non-user friendly hand gesture.

The number of key-presses and finger travel is almost similar for Melo and Half-qwerty keyboard. But for typing in Melo, the fingers move together (with very less or no stretching of fingers). Where as, in Half-qwerty keyboard, two characters of roman script are mapped on one key, i.e.. Four characters of Devanagari are mapped on one key. Typing in half-qwerty keyboard also requires awkward and non-user friendly hand gesture.
For e.g.. To type ‘आ’ in Half-qwerty, Flip key is pressed simultaneously with the shift key and key with ‘आ’ (flip + shift + e).

The following tables (table 2,3 & 4) give an overview of the comparison between Melo, InScript and Half-qwerty keyboard for different set of words (based on the complexity of their character-combinations).

FOR SET 1

	MELO	INSCRIPT	HALF-QWERTY
KEY PRESS	39	22	37
FINGER TRAVEL	78	118	88

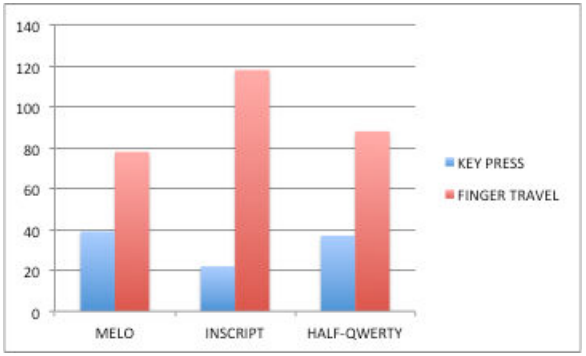


Table 2: Comparison for no. of key presses and finger travel
For Set 1 : Easy level-
कमल पवन भारत रसिका और मित्र हैं

FOR SET 2

	MELO	INSCRIPT	HALF-QWERTY
KEY PRESS	44	29	38
FINGER TRAVEL	93	149	82

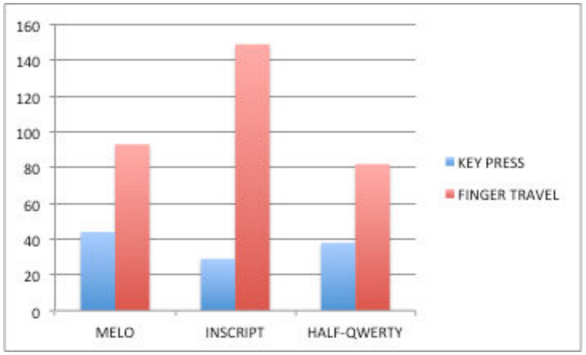


Table 3: Comparison for no. of key presses and finger travel.
For Set 2 : Medium level-
गणपति स्टार प्रयास क्यूँ स्थान

FOR SET 3

	MELO	INSCRIPT	HALF-QWERTY
KEY PRESS	41	34	46
FINGER TRAVEL	86	135	88

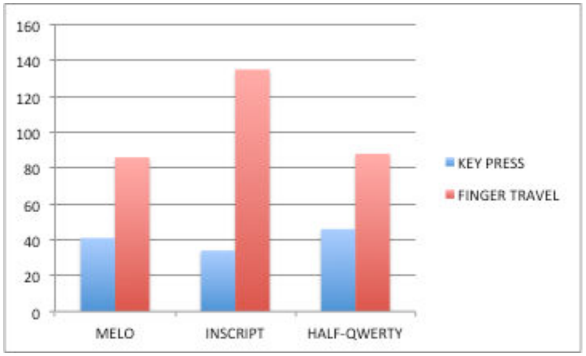
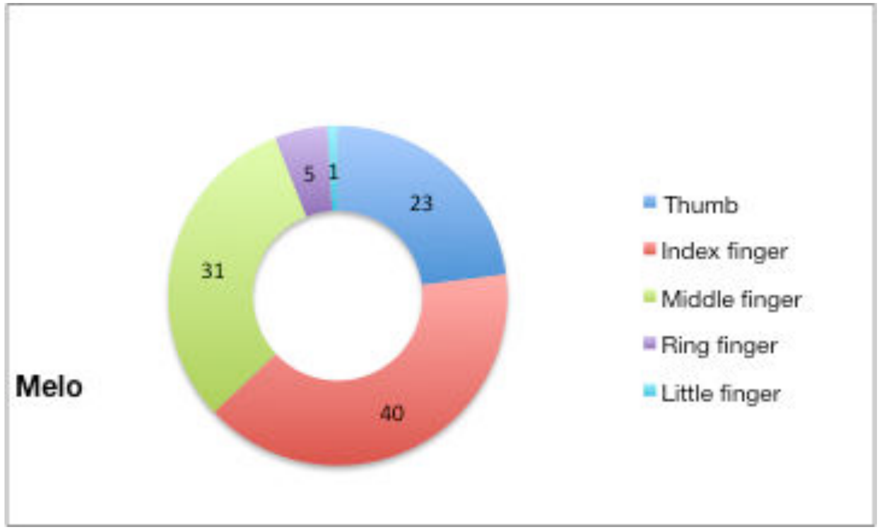
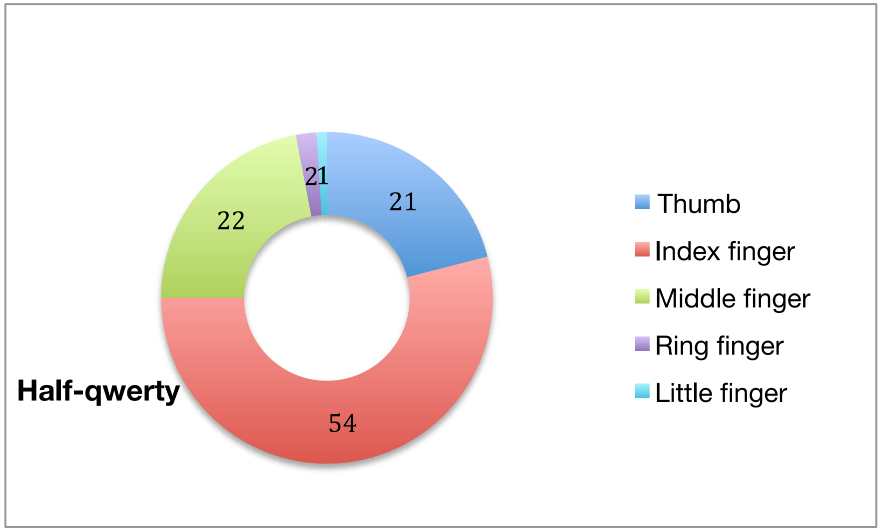


Table 4: Comparison for no. of key presses and finger travel.
For Set 3: Difficult level-
शर्मा क्षत्रिय आर्य त्रिशंकु कृष्णा

2. Finger Load





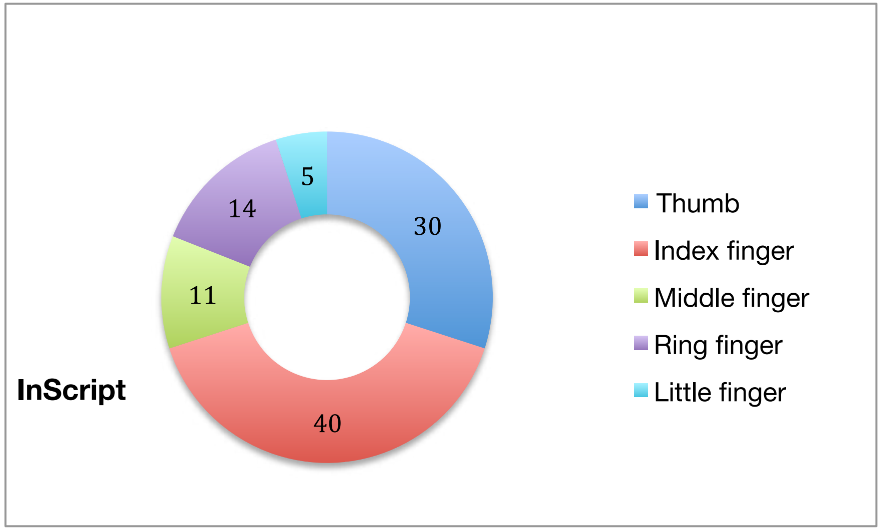


Figure 64 (a, b, c): Finger load on different fingers for Melo, InScript and Half-Qwerty Keyboard.

The pie charts (Fig. 64) give an overview about the distribution of load on various fingers for the three keyboards.
In **Melo**, the load is distributed majorly on the fingers with higher strength and pointing accuracy (Load on Index finger> Ring finger > Thumb). Also, the stretching gesture of fingers being least (as compared to InScript and half-Qwerty) makes it user-friendlier.
In **Half-Qwerty**, the load is distributed mainly on index, middle finger and thumb, but there is excessive load on ring finger. Also, typing in Devanagari using a Half-qwerty keyboard requires awkward and non-user friendly hand gestures (as mentioned in Section 8.2.1b-1).
In **InScript**, typing with one hand requires involvement of fingers with lesser strength and pointing accuracy (Ring and little finger).

8.2.2 **Empirical evaluation**

Two different sets of 6 users each were recruited (ref. to User Criteria in section 8.1) for conducting the Short term testing on Melo and InScript Keyboard (when used with One-hand). 4 users out of the set of 6 users were further recruited for Long-term testing on the same keyboard that they used while Short-term testing. The Methodology is discussed as under-

8.2.2a **Methodology for-Short term testing (Melo v/s InScript)**

1. A 15-minute demonstration/walk-through of the Keyboard, which includes-
    a. Familiarizing the user to the prototype.
    b. Demonstrating the user to type 5 words having characters that are formed by different combinations such as C+V, C+C+V etc.
    c. Demonstrating the user the difference in cognitive style of typing/ speaking as compared to that of writing.
    d. Demonstrating the user on how to put halant and other special matras such as rafar, rashtra chinh etc.

2. Allowing the user to use the prototype and do self-exploration for about 15 minutes.

3. The users were made to type the three set of words in three sessions with 10-minute break in between. Time is recorded for each set of words.

Following are the three set of words-

SET 1: EASY
पवन अगर कमल कविता रसिका नदी

SET 2: MEDIUM
मैं चाँद अंकित व्यास स्टार मस्त

SET 3: DIFFICULT
अर्थात् कृष्णा क्यूँ प्राण श्रेयासी व्यक्ति

8.2.2b **Methodology for-Long term testing (Melo v/s InScript)**

After first day of Short term testing, long term testing was conducted from second day onwards. 4 sessions were conducted with 1-day break in between. In every session, the user has to type 50 words from 7th standard Hindi textbook. Time and number of errors was recorded for every session.

8.2.2c **Goal of Empirical Evaluation-**

The goal of Empirical evaluation was to compare the effectiveness of Melo Keyboard as compared to the InScript keyboard for the following factors-

1. Learnability for the first-time user

2. Typing Speed

3. Occurrence of errors

Figure 64: Users testing the Melo Keyboard and InScript Keyboard

8.2.2d **Results of Empirical Evaluation-**

SHORT TERM STUDY

Time taken (in seconds)

INSCRIPT

USERS	EASY	MEDIUM	DIFFICULT		OVERALL
1	81	111	139		331
2	70	115	192		377
3	136	163	220		519
4	104	141	172		417
5	98	125	158		381
6	85	92	170		347
AVERAGE	96	125	175		395

MELO

USERS	EASY	MEDIUM	DIFFICULT		OVERALL
1	80	90	118		288
2	78	128	156		362
3	139	159	160		458
4	118	140	150		408
5	95	115	131		341
6	91	104	134		329
AVERAGE	100	123	142		364

COMPARISION

	EASY	MEDIUM	DIFFICULT	OVERALL
INSCRIPT	96	125	175	395
MELO	100	123	142	364

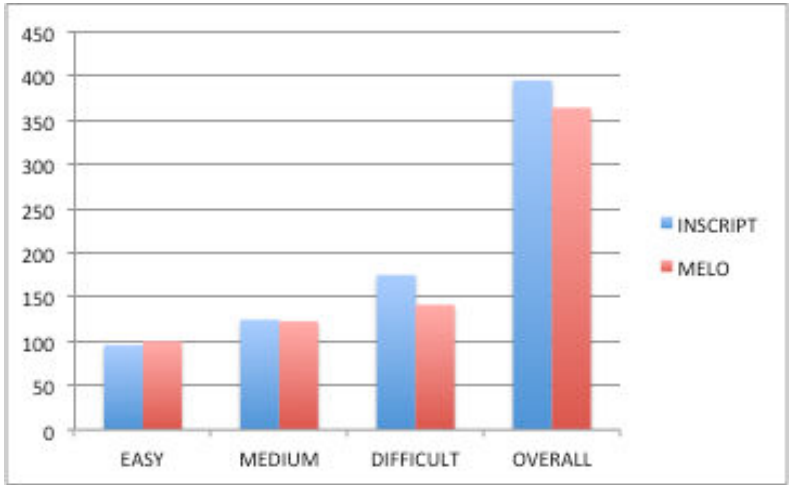


Table 5: Results of the short-term study

LONG TERM STUDY

FOR **INSCRIPT**

Time taken in each session (50 words per session)

Time (in seconds)

	USER 1		USER 2		USER 3		USER 4			TOTAL	
	Time taken	No. of errors	Time taken	No. of errors	Time taken	No. of errors	Time taken	No. of errors		Time taken	No. of errors
SESSION 1	614	4	719	4	821	3	867	1		3021	12
SESSION 2	479	3	631	3	663	2	675	0		2448	8
SESSION 3	491	4	565	3	657	1	596	3		2309	11
SESSION 4	557	3	699	3	605	2	610	2		2471	10
	2141		2614		2746		2748		**TOTAL**	10249	41

FOR **MELO**

Time taken in each session (50 words per session)

Time (in seconds)

	USER 1		USER 2		USER 3		USER 4			TOTAL	
	Time taken	No. of errors	Time taken	No. of errors	Time taken	No. of errors	Time taken	No. of errors		Time taken	No. of errors
SESSION 1	520	9	693	5	861	6	678	3		2752	23
SESSION 2	397	4	581	4	642	4	563	2		2183	14
SESSION 3	393	4	459	3	542	5	511	2		1905	14
SESSION 4	512	1	534	2	508	2	552	2		2106	7
									TOTAL	8946	58

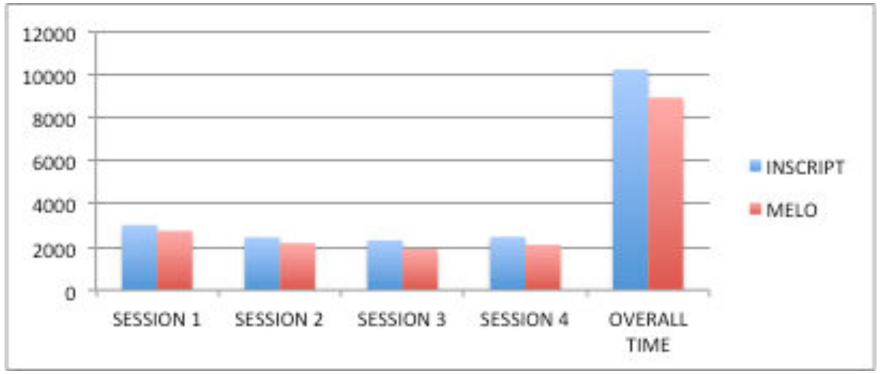
Table 5: Data sheet of the Long-term study for Inscript and Melo.

LONG TERM STUDY- COMPARISION

TIME COMPARISION

Time (in seconds)

	INSCRIPT	MELO
SESSION 1	3021	2752
SESSION 2	2448	2183
SESSION 3	2309	1905
SESSION 4	2471	2106
OVERALL TIME	10249	8946
AVERAGE TIME (per user)	2562	2237



ERROR COMPARISION

	INSCRIPT	MELO
SESSION 1	12	23
SESSION 2	8	14
SESSION 3	11	14
SESSION 4	10	7
TOTAL NO. OF ERRORS	41	58

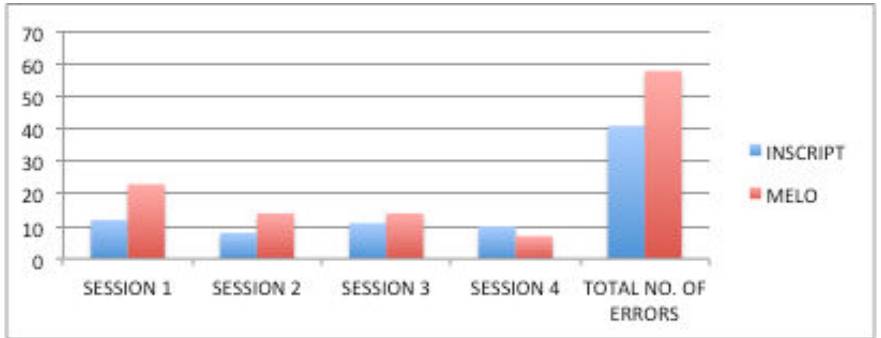


Table 6 (a, b): Comparison of the Data sheet of the Long-term study

8.2.2d **Results of Empirical Evaluation-**

1. Learnability for the first-time user-

The observations during the short-term study and the user feedback after the test made it vivid that the first time users found Melo keyboard easy and intuitive to learn. By the time they finished short-term test, they were able to develop the conceptual model of the keyboard. While using Melo keyboard, the users found it difficult to use initially as the conceptual model was difficult to develop for them and as a result, it resulted in more fundamental errors in typing the 1st set. The panel placed over the keyboard was found helpful (Figure 55 & 61) in developing the conceptual model.

Based on the observations, the users referred to the information panel on the Melo keyboard for every letter they typed initially. It was reduced to a great extent by the time they reached 3rd set; as by then, the users developed their own conceptual model. For e.g.. One user told that for certain frequently occurring words, he remembered the keys required to input them. For e.g.. To input 'स', 7th and 8th white key are pressed simultaneously and so on.

While using the InScript keyboard, the users had to hunt and peck for the characters. By the end of the short-term study, they too were able to develop the conceptual model for the InScript keyboard. Because the letters were marked on the keys in InScript, it required lesser cognitive load for the users to learn it, which enhanced the ease of learning.

The results of the short-term study suggest that the overall time required in completing the 3 sets of words was less in Melo as compared to InScript. Based on t-Test for Two-Sample assuming Equal Variances (for short-term study), the difference in overall time on Melo as compared to InScript was not significant (p= 0.4217). The reason could be smaller sample size (No. of users= 6). But, out of 3 different sets of words, the difference in time for 3rd set for Melo as compared to InScript was significant (p= 0.0294). But, it is insufficient to draw a conclusion.

2. Typing speed

For measuring the typing speed, the data gathered from the long-term study (table 6a) is considered. For all the sessions, the typing speed on Melo was 12.7% better than InScript keyboard (used with one-hand). In session 1, 2, 3 and 4, the typing speed in Melo was 8.9%, 4.9%, 17.7% and 14.7% better than in InScript keyboard respectively.

Based on t-Test for Two-Sample assuming Equal Variances (for long-term study), the overall difference in typing speed is not significant (p-0.7805). The difference is not significant for all the individual sessions except in session 4 (p=0.0268), which is insufficient to draw a conclusion. A larger sample size would have provided better results.

3. Occurrence of Errors

For measuring the occurrence of errors, the data gathered from the long-term study (table 6b) is considered. In the 1st session, overall 23 errors occurred in Melo as compared to 12 errors in InScript. Familiarity with prototype of Melo Keyboard is required, as it requires sequential and simultaneous key presses for typing. The users were finding it difficult in the 1st session to perform the required key-presses. Also, there were certain limitations of the prototype's hardware that came into play (discussed in section 7.2.2).

After the 1st session, the number of errors dropped in Melo Keyboard and by 4th session the numbers of errors were almost similar to that of InScript.

Based on the t-Test for Two-Sample assuming Equal Variances (for long-term study), the difference in errors for InScript and Melo Keyboard was not significant (p=0.2563).

8.3 Future Scope

1. **Statistical analysis of the evaluation data**
   T-Test for Two-Sample (assuming Equal Variances) was conducted, but the results were not significant enough to draw conclusions. The empirical evaluation should be conducted on a larger sample size for better results and conclusion.

2. **Usability Evaluation**
   Even though a long-term evaluation with 200 words (in 4 sessions) was conducted on Melo and InScript, the results were not reliable as the sample size was small (4 users). A detailed longitudinal usability evaluation is required to compare the performance of Melo Keyboard with InScript Keyboard (for one-handed use).
   The evaluation that was conducted was very generic as the various parameters were loosely controlled. Appropriate tests must be designed for evaluation to come up with accurate results. The concepts must be compared on the following criteria (with more control on the parameters)-
   - Ease to learn and Ease to use
   - Speed of use (characters per second)
   - Errors (Errors per character)

3. **Product design**
   As a scope of this project, only a basic method to input text is designed. No emphasis was laid on the form and aesthetics of the product. Being a physical keyboard, the form of the product needs to be designed.

4. **Language Prediction**
   Language prediction model can be used to predict the entire word or next probable letters, which would in turn increase the efficiency of the input system.

5. **Introduction of multiple modal inputs**
   Various possibilities can be explored by integrating multimodal inputs with the Melo keyboard.

9

References

1 I. Scott MacKenzie, William Buxton Edgar Matias. (1993) matias.ca. [Online]. www.matias.ca/halfkeyboard

2 I. Scott MacKenzie, William Buxton Edgar Matias, "One-Handed Touch-Typing on a QWERTY Keyboard," in *INTERCHI '93*, 1993.

3 Wikipedia. [Online]. http://en.wikipedia.org/wiki/Devanagari

4 Richard Salomon, "Writing system in Indo-Aryan languages," 2003.

5 Ashish Ganu, Aditya Chand, Vikram Parmar, Gaurav mathur Anirudha Joshi, "Keylekh: A Keyboard for Text Entry in Indic Scripts," in *CHI 2004*, Vienna, 2004.

6 Hemruchi Shah, Anirudha Joshi Anshuman Kumar, "Saral: Devanagari Text input system in Mobile phones," in *Indo-Danish HCI Research Symposium*, 2006.

7 Aniket Mohan Sarangdhar, "Devanagari Text Input- Touch screen Mobile device," Industrial Design Centre, IIT Bombay, Mumbai, 2010.

8 Manoj Kumar Sharma, "Word prediction system with virtual keyboard for text entry in Hindi," School of Information technology, IIT Kharagpur, Kharagpur, 2012.

9 Patrick Brooks, Hope Harley, Mike McGee Brian Amento. (1996) Fitt's Law. [Online]. http://ei.cs.vt.edu/~cs5724/g1/

10 Anastasios Karafillis. (2012, December) Smashing Magazine. [Online]. http://uxdesign.smashingmagazine.com/2012/12/04/fittss-law-and-user-experience/

11 Cykey Keyborad-Microwriting. [Online]. http://www.cykey.co.uk/

12 Wikipedia. [Online]. http://en.wikipedia.org/wiki/QWERTY

13 (2006) The Qwerty Keyboard Layout Vs The Dvorak Keyboard Layout. [Online]. http://www.andong.co.uk/dvorak/

14 microsoft.com. [Online]. http://www.microsoft.com/enable/products/altkeyboard.aspx

15 Jan Noyes, "The Qwerty Keyborad: A review," *International Journal of Man-Machine Studies*, 1983.

16 Wikipedia. [Online]. http://en.wikipedia.org/wiki/Chorded_keyboard

17 Computing History. [Online]. http://www.computinghistory.org.uk/det/5798/Microwriter-new-users-guide/

18 Frog Pad- Wikipedia. [Online]. http://en.wikipedia.org/wiki/FrogPad

19 Frog Pad-website. [Online]. http://www.frogpad.com/

20 Digital501.com [Online]. www.digital501.com

21 Inscript-Wiki. [Online]. http://en.wikipedia.org/wiki/InScript_keyboard

20 Digital501.com [Online]. www.digital501.com

21 Inscript-Wiki. [Online]. http://en.wikipedia.org/wiki/InScript_keyboard

22 Girish Dalvi, Manjiri, Prasad, Aniket Sarangdhar Anirudha Joshi, "Design and Evaluation of Devanagari Virtual Keyboards for Touch Screen Mobile Phones," in *MobileHCI 2011*, Stockholm, 2011.

24 Wikipedia. [Online]. http://en.wikipedia.org/wiki/Abugida

25 Wikipedia. [Online]. http://en.wikipedia.org/wiki/InScript_keyboard