

“T-Board: The Total Keyboard” For M-Learning Platform

**Principal Author: Mr. Vishwas Kulkarni, Computer Home Private Limited,
Pune.(www.computer-home.com). Co-Authors: Mr. Kishor Deshmukh, Ms. Puja Pathak, Ms.
Gauri Pathak, Mr. Amol Khandhare.**

Abstract -The M-learning is new emerging platform to present and deliver contents based knowledge using mobile/tablet devices. But the most difficult task is to prepare mathematical and scientific contents on those devices, as these devices do have very small screens and keyboards and modern authors do prefer mobile/tablets for content preparation over laptops/desktops because of mobility. Unfortunately there are no applications on tablet/mobile platforms for content preparations. However desktop/laptop computers have lot of applications for content preparations.

Our research was primary aimed to find innovative ways to prepare contents quickly by providing context specific keyboards as well as switching these keyboards quickly with press of easily locatable key on the existing keyboard. Our research was also focused to find innovative ways to present the content as per various MS-OFFICE formatting and style options by providing MS keyboard.

After finding these innovative ways, we have written “T-Board” application on Android Platform and given to various authors for their review and suggestions. After almost 10 iterations and revisions, our authors are satisfied and started preparing scientific and mathematical contents on their tablet/mobile platforms. These authors have also expressed us that “T-board” made their life very easy as far as contents preparation on mobile platforms. They also expressed that their productivity has increased many-fold in our 10th revision when compared to our 1st version.

So our final version (release 10) “T-board” solves the issue of content preparation on small devices by providing context specific keyboards which are fast switchable. For example, if you wish to prepare the notes for the topic Trigonometry in Geometry subject using smart phone. Then you will require specific symbols like λ , ϕ , α , degree, $\wedge 2$ etc. which will be provided by our context specific keyboard i.e. “T-Board”.

Even “T-Board” cater some innovative ways in preparing HTML based content using presentation keyboard.

Index Terms – Android, Smart Phones, Keyboard, T-Board.

I. INTRODUCTION

The product perspective is to provide the teachers/authors the context specific keyboards with fast switching mechanism so that more and more contents will be prepared on mobile/tablet platform with ease and pace. According to the

survey it has been noticed that if teachers/authors wish to prepare notes in any platform such as word then adding formulas, symbols becomes difficult for them. This problem is effectively addressed by the “T-Board” project. The “T-Board” also provides easy and innovative ways to prepare HTML based presentation content.

In short, the “T-Board” provides:

- (1) Context specific keyboard with fast switching among various keyboards.
- (2) Easy and innovative ways to enter various types of scientific and mathematical formulas and
- (3) Easy, user-friendly ways to prepare HTML based presentation of the content.

It also solves ergonomics related issues like continuous finger movements and health hazards caused because of the same.

Manuscript April 14, 2014; Revised version April 14, 2014.

This work was supported by COMPUTER HOME PVT. LTD in terms of its objectives ideas and technical skeleton and methodologies, testing and guidance. Project Number CHPL-2013-2014(2), Project title: T-Board: The Total Keyboard.

Project Guide and Principal Author: Vishwas Kulkarni -Project Guide from Computer Home Private Limited, Pune, India.

First Co-Author name: K. Deshmukh is studying in Govt. College Of Engineering And Research, Awasari(kd), Pune.

Second Co-Author name: P. Pathak is studying in Govt. College Of Engineering And Research, Awasari (kd), Pune.

Third Co-Author name: G. Pathak is studying in Govt. College Of Engineering And Research, Awasari (kd), Pune.

Fourth Co-Author name: A. Khandare is studying in Govt. College Of Engineering And Research, Awasari (kd), Pune.

II. LITERATURE SURVEY OF THE PROJECT

Any SoftKeyboard:

- Less special Symbols.
- No sorting of Symbols.
- No HTML support.
- Lightweight GUI.

GO Keyboard:

- Navigation for text provided. But for HOME and END keys, the cursor goes to HOME/ END of the whole text, instead of HOME/ END of current line.
- Symbols provided are sorted.
- No HTML support.
- Heavy GUI.

E-learning has already started, then why for M-learning?

The primary differences between the two disciplines are the distinction between more text- and graphics-based instructions to more voice-, graphics-, and animation-based instructions. Where learning previously occurred in front of a computer terminal, in the classroom, laboratory, or at home, it is now enabled to occur in the field, or at any location where the mobile[3] device is fully functional.

III. CREATING COMPLEX MATHEMATICAL AND SCIENTIFIC EXPRESSIONS FOR M-LEARNING

We addressed the problems of existing keyboards for m-learning platform and following are the requirements:-

Design context specific keyboard (Subject and Topic as context) consisting only mathematical and scientific symbols required for the context. Provide easy and innovative “**One Key Stroke Structure**” [2] for entering various formulas like sin(), cos(), ∫ with its limits. i.e. user has to press only one key and the structure of related key will be provided in Edit Textbox.

e.g.

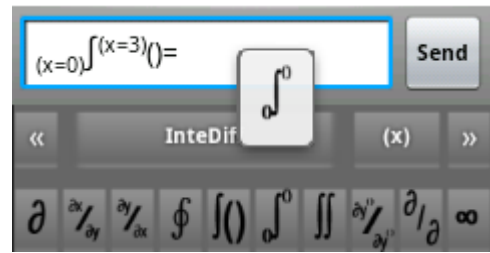


Fig.1. On pressing ∫ key.

Other examples of One Key Structure can be seen in Fig. 6 such as Σ, lim, log, etc. To provide easy and innovative “**One Key Stroke Structure**” for MS-Word like presentation formats like **Bold**, *Italics*, Underline, etc..

e.g.



Fig.2. Various MsWord’s features.

Proper representation of Numerator & Denominator as shown in Fig. 3

e.g.

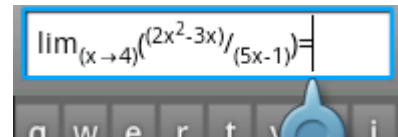


Fig.3. N^r & D^r representation.

IV. PROPOSED SYSTEM

For faster creation of required content the keyboards should have grouped keys and easy switching between the keyboards. Thus we’ve grouped the keyboard by chapter-names.

For some functions like sin(), cos(), tan(), [1] etc., after clicking on the button, the cursor should be set in the bracket. As shown in Fig. 4.

e.g.

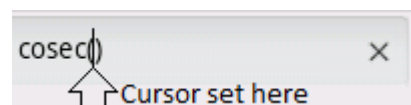


Fig.4. Setting the cursor automatically on right place

And the code for the same can be written as:

For cosec():

```
getCurrentInputConnection().commitText("cosec()",0);
```

And for setting the cursor:

```
getCurrentInputConnection().commitText("",-1);
```

For superscript we will be direct rendering the output via HTML code and put it in the current EditText. But as this HTML code works only in tag, so we will be providing the tags with bracket contained. This everything written in that bracket will get the properties of that HTML tag.

e.g.

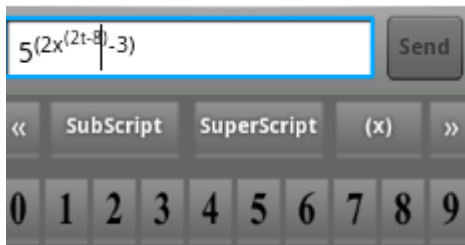


Fig.5. Text within tags written in brackets.

Code:

```
getCurrentInputConnection().commitText(
Html.fromHtml("<sup></small>()</small></sup>"),0);
```

Here the superscript content can be written in those brackets. As shown in the Figure 5. we can form a hierarchy of superscripts/subscripts by putting brackets inside it recursively. And we can write any expressions in superscript. Thus, making it easy to write even more complex expressions.

The limits of integration, N^r , D^r , summation, $\lim_{x \rightarrow 0}$, etc. the representation can be done in superscript/subscript as,

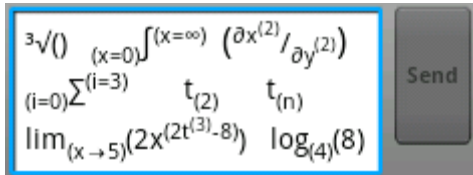


Fig.6. Various combinations using Superscript/Subscript

And many more vivid combinations can be tried by using superscript/subscript. The code for $\lim_{x \rightarrow 0}$ & $\log_{(0)}$, where “|” is cursor, would be:

case LIMIT:

```
getCurrentInputConnection().commitText(Html.from
Html("<lim<sub><small>(x&rarr;)</small></sub>()="),0
);
getCurrentInputConnection().commitText("",-4);
break;
```

case LOG:

```
getCurrentInputConnection().commitText(Html.from
Html("<log<sub><small></small></sub>()"),0);
getCurrentInputConnection().commitText("",-1);
break;
```

case DIVISION_NrDr :

```
getCurrentInputConnection().commitText(Html.from
Html("<sup><small></small></sup></sub><small></small></sup>"),0);
getCurrentInputConnection().commitText("",-5);
break;
```

Where the cursor will be set to the appropriate place.

Similarly the code of features of MS-Word as shown in Fig 2. is:

case BOLD :

```
getCurrentInputConnection().commitText(Html.from
Html("<b></b>"),0);
getCurrentInputConnection().commitText("",-1);
break;
```

case ITALIC :

```
getCurrentInputConnection().commitText(Html.from
Html("<i></i>"),0);
getCurrentInputConnection().commitText("",-1);
break;
```

case UNDERLINE :

```
getCurrentInputConnection().commitText(Html.from
Html("<u></u>"),0);
getCurrentInputConnection().commitText("",-1);
break;
```

case GREEN :

```
getCurrentInputConnection().commitText(Html.from
Html("<font color='green'></font>"),0);
getCurrentInputConnection().commitText("",-1);
```

break;

Thus many more functions like other colors, bullets, etc. can be implemented using HTML rendering.

Now as there are many symbols they need to be placed and properly grouped. This will be beneficial for the user to find the required key quickly. This can be done by popping the similar keys on long pressing a specific key.

For e.g.

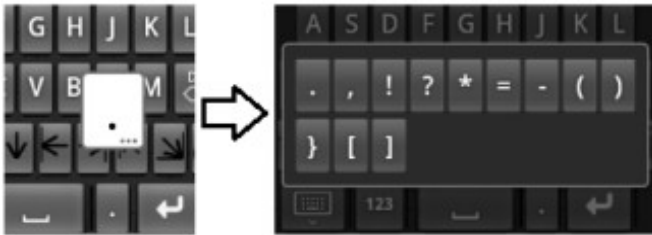


Fig.7. Popup Keyboard on long key press.

On pressing a key the hint about which key is being pressed currently is given by positioning a white hovering key. For a normal key this hovering key has a plain white background. But for multiple keys, the background consists of three black dots at bottom. The code for this is to be placed in XML file. And specific keys or even a whole keyboard can be popped up like this. Only the drawback here is no repeatable keys allowed here, only one time press from popped up keyboard are possible.

V. TECHNICAL ISSUES

The text written in EditText are actually output of HTML code. Thus, while retrieving of the text the content becomes plain text .e.g.

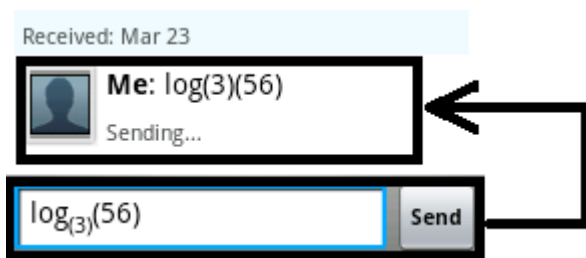


Fig.8. Text becomes plain if not processed properly.

Here consider a chat application in which, after sending the text in HTML as SMS, the text converts in plain text. The same thing happens with saving of the text in a “.txt” extension file on external/internal SD-Card.

Also every time after writing superscripts, subscripts, colors, etc. they are always to be written in opening and closing round brackets. So, if the user wants to remove the brackets he has to set the cursor at appropriate position and

then delete them. On many android devices no hardware D-PAD keys are given for quick navigation in text. Thus, this can be a very time consuming process. In internal keyboard there is no existence of few useful keys like HOME, END, COPY, PASTE, Navigation keys, etc.

VI. TECHNICAL SOLUTIONS

The saving of the HTML output should be done with no data loss. Thus by importing the HTML class files, we can use its functions to extract the text as HTML. This kind of text extraction can extract almost all the tags except, the `<big></big>` and `<small></small>` tags. Thus these two tags should be placed with colors in HEX like black with a very minor variation in BITS.

Code for Big and Small tags (an issue):

case BIG :

```

getCurrentInputConnection().commitText(Html.from
Html("<big>()</big>"),0);
getCurrentInputConnection().
commitText("",-1);
break;

```

case SMALL :

```

getCurrentInputConnection().commitText(Html.from
Html("<small>()</small>"),0);
getCurrentInputConnection().
commitText("",-1);
break;

```

Output for this code:

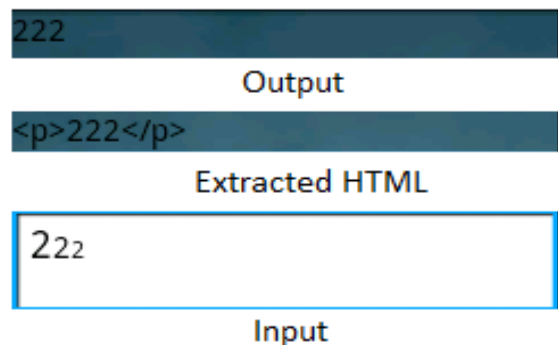


Fig.9. Output on processing the input incorrectly.

Here the number two is arranged in order Big, Normal and Small, for comparison. Code for Big and Small (solution in T-Board):

case BIG :

```
getCurrentInputConnection().commitText
(Html.fromHtml("<big><font color= '#000001'>
()</font></big>"),0);
```

```
getCurrentInputConnection().commitText("",-1);
break;
```

case SMALL :

```
getCurrentInputConnection().commitText(Html.from
Html("<small><font
color='#000000'>()</font></small>"),0);
```

```
getCurrentInputConnection().
commitText("",-1);
```

break;

A sample of colored text and Big and Small text is given below:

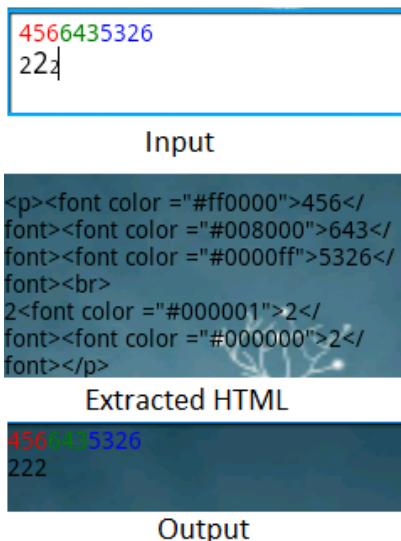


Fig.10. Output on processing the input of only colored text correctly.

In the app above application input is taken in Edit Text Box, which is being extracted as HTML. Here the input consists of HTML output of Big and Small tags being correctly displayed. But in the extracted text as HTML, the <big></big> and <small></small> tags get missing. Also in Figure 10. we get the output of Big and Small as plain normal text. But the font is correctly extracted, making no big difference in the colors of output formed.

Thus if the text extracted is replaced by <big></big> and <small></small> tags by help of these color tags, we can preserve the Big and Small tags. Then the output is shown in the following figure:

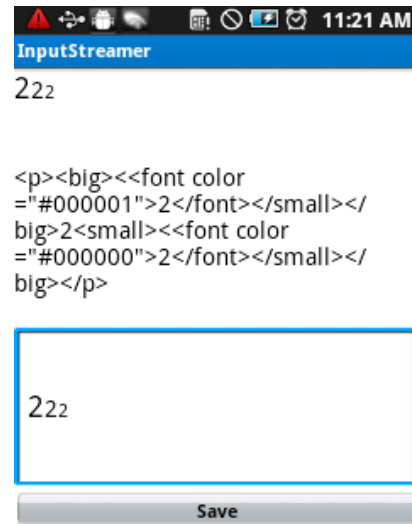


Fig.11. Output on processing the input of Big and Small tags correctly.

The code for which is given below (solution in your application):

```
str=""+Html.toHtml(et_get.getText());
if(str.contains("<font color=\\'#000001\\'>"))
{
Log.d("HTML___if", " str: "+str);
str=""+str.replaceAll("<fontcolor=\\'#000001\\'>","<big
><font color = \\'#000001\\'>");
str=""+str.replaceAll("</font>","</font></big>"); }
if(str.contains("<font color=\\'#000000\\'>"))
{
Log.d("HTML___if", " str: "+str);
str=""+str.replaceAll("<font color= \\'#000000\\'>","
<small><font color =\\'#000000\\'>");
str=""+str.replaceAll("</font>","</font></small>"); }
Log.d("HTML___after___if", " str: "+str);
tv_plain.setText(Html.fromHtml(str));
```

V. APPLICATIONS

The T-Board is used in Quiz Authoring & Quiz Taking Applications of Computer Home Pvt. Ltd. These applications are made for mobile-learning platform for creating quizzes or notes of any subject for e.g. Math, Physics, Chemistry, etc. Following snapshots represents the short demo of the T-Board used in Quiz Authoring (for teachers) & Quiz Taking Applications (for students):

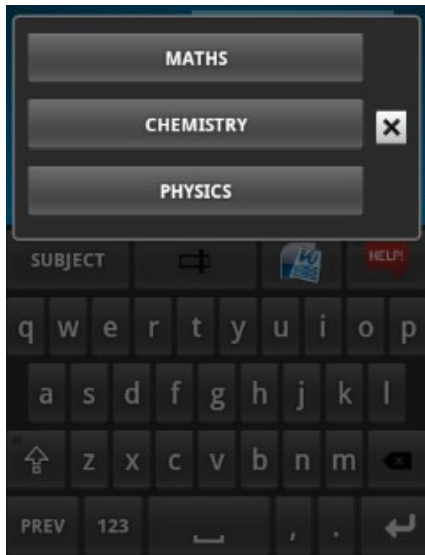


Fig.12. Subject popping-up as a keyboard on the long press of Subject key



Fig.13. Chapters popping-up as a keyboard on the long press of Chapter key

For faster navigation through the keyboards, the keyboard titles are popped up as a keyboard and the required keyboard can be selected from the list.

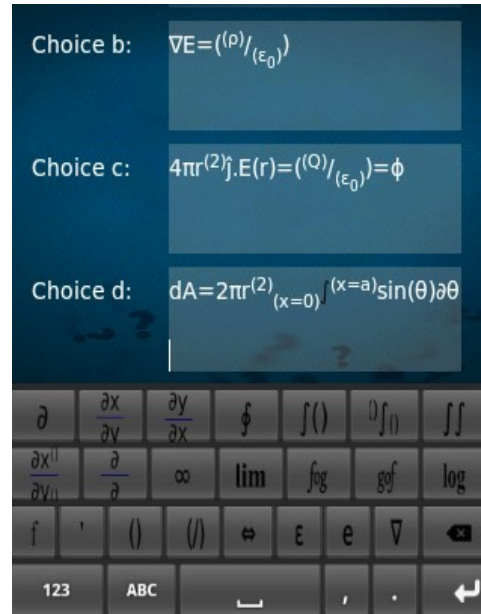


Fig.14. Integration-Differentiation Keyboard in Quiz Authoring Application. The expressions are saved on SD card by extracting the text as HTML itself.

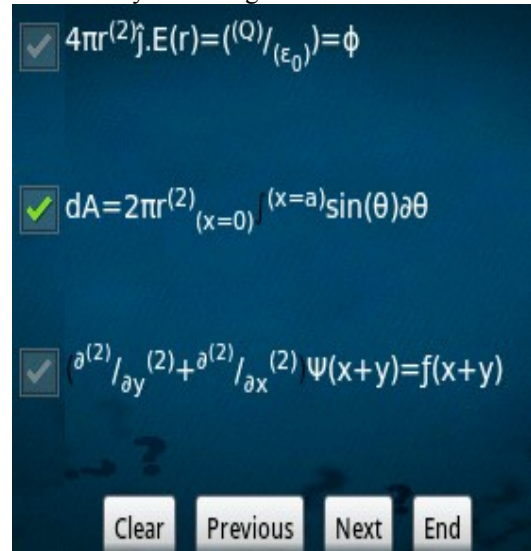


Fig.15. Representing Quiz taking application where the expression saved are retrieved as it is from SD Card possible because of HTML Rendering



Fig.16. Representation of Chemistry Expression using Chemistry Keyboard

In Chemistry keyboard, symbols right from the Thermal energy to standard chemical reactions can be typed where phenol is a single character.

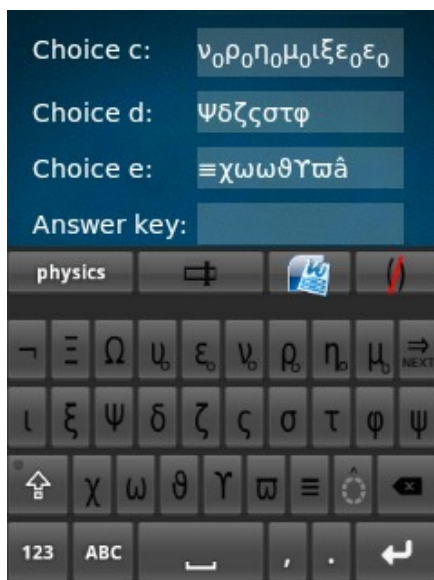


Fig.17. Physics Keyboard

The Physics keyboard, in which relevant symbols are placed including the vector cap for letters and the added zero to the subscript.



Fig.18. MSWORD Keyboard

In MSWORD keyboard, the representation of different font style such as Italic, Bold, Underline, Bigger and Smaller text, Color text using the functions of MS Word. Also 2*2 and 3*3 matrix are provided.



Fig.19. Navigation Keyboard

In Navigation keyboard, there are some extra keys:

- (a) THOME (Total Home): Sets the cursor at the end position of the current document.
- (b) TEND (Total End): Sets the cursor at the start of the current document.
- (c) HOME: Sets the cursor at the start of the current line.
- (d) END: Sets the cursor at the end of the current line.
- (e) SELECT: For the selection of text.

The FROYO and later versions of Android devices lack the physical keys for D-PAD. So the navigation keys can be often useful for navigating through text.

And Research, Awasari www.gcoeara.com)

VI. CONCLUSION

This way you can make your application support T-Board. And thus making convenient for the teachers/authors for preparing notes more efficiently.

VIII. REFERENCES

1. R. M. ElSheikh and S. D. Najdi, "Math Keyboard Symbols and its Effect in Improving Communication in Math Virtual Classes".
2. Shumin Zhai, Michael Hunter, and Barton A. Smith IBM Almaden Research Center, "Performance Optimization of Virtual Keyboards"
3. Assist. Prof. Dr. Santosh Kumar BEHERE (Department of Education Sidho-Kanho-Birsha University Purulia, West Bengal, India)"E-Learning and M-Learning: A Comparative Study"

Guided and Sponsored By:-



Mr. Vishwas Kulkarni

(Director of Computer Home Private Limited, Pune www.computer-home.com)

Note: "T-Board" Application software Intellectual Property rights belongs to "COMPUTER HOME PRIVATE LIMITED, PUNE, INDIA".



Mrs. Pratiksha Deshmukh

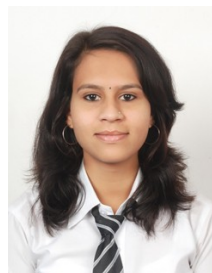
Assistant Professor (Government College of Engineering



Mr. Kishor P. Deshmukh

Pursuing B.E. (Computer Engineering)

University of Pune, Department of Computer Engineering, Government College of Engineering and Research Awasari (Khurd), Taluka- Ambegaon, Pune.



Ms. Puja A. Pathak

B.E. (Computer Engineering)

University of Pune, Department of Computer Engineering, Government College of Engineering and Research Awasari (Khurd), Taluka- Ambegaon, Pune.



Ms. Gouri A. Pathak

B.E. (Computer Engineering)

University of Pune, Department of Computer Engineering,
Government College of Engineering and Research Awasari
(Khurd), Taluka- Ambegaon, Pune.



Mr. Amol S. Khandare

B.E. (Computer Engineering)

University of Pune, Department of Computer Engineering,
Government College of Engineering and Research Awasari
(Khurd), Taluka- Ambegaon, Pune.