Virtual keyboard with scanning and augmented by prediction

Peter E Jones

Electrical and Electronic Engineering, The University of Western Australia,
Nedlands WA 6907, Western AUSTRALIA

peterj@ee.uwa.edu.au

ABSTRACT

All our teenage users are confined to electric wheelchairs and are unable to speak or make any voluntary movements much beyond either moving their head against one of three switches mounted in the chair's headrest or to hit a large "banger" switch. Real-world devices are beyond reach, only devices in a virtual world are attainable. This virtual keyboard project was designed to meet their needs for interacting with commercial off-the-shelf software such as word processors, spreadsheets, electronic mail and Internet tools. The virtual keyboard uses scanning augmented by character prediction and word completion.

1. INTRODUCTION

Children who have disabilities have reduced opportunities in all areas, more especially in education and in social settings (Alm et al 1992). The group we are working with are physically dependent and cannot control their environment. Often this can cause a problem referred to as "learned helplessness." Virtual Reality (Singh et al 1996) offers the user who, because of their impairment, cannot control or operate in the real world but can do so in a virtual one. Ellis (1995) defines virtualisation as the process by which a human viewer interprets a patterned sensory impression to represent an extended object in an environment other than that in which it physically exists. In his most abstract version it requires users to interpret an image on a flat screen with many of the usual cues missing as representing a constructed object. While this project does not provide an immersive Virtual Reality, it does provide a virtual device with which the children can interact. For them, it is a virtual world where they can type on a keyboard. Having the ability to produce text opens further doors in communication and information access without the need for one-on-one intervention from a teacher or therapist.

2. PROJECT'S AIMS

From our observations (Jones, 1997) of the users with our earlier projects we realised that activating one switch was very time consuming and error prone. To interact with a word processor effectively we would need to find some means of speeding up the interaction. We are not alone in tackling this problem, for example the Reactive Keyboard (Darragh & Witten, 1992), work at the University of Dundee in the UK (Hine et al, 1994) and a number of papers in Edwards (1995) provide solutions. Commercial systems exist as well, for example SofType (Origin Instruments, 1998). However, they don't address issues beyond those concerned with prediction or are not adaptable enough or cannot meet our users' needs.

Our prototype, developed in Microsoft Visual Basic, is shown in Figure 1. If this were to be shown in animation and colour, then you would notice the moving cursor down the left-hand column indicating the current scan row. The vowels are shown in red, the numbers in a purple colour and special keys shown in blue. Other characters are shown in black. At the end of each row is our innovative "go back" symbol (shown as a "<") which, if selected, repeats the scan of that row.

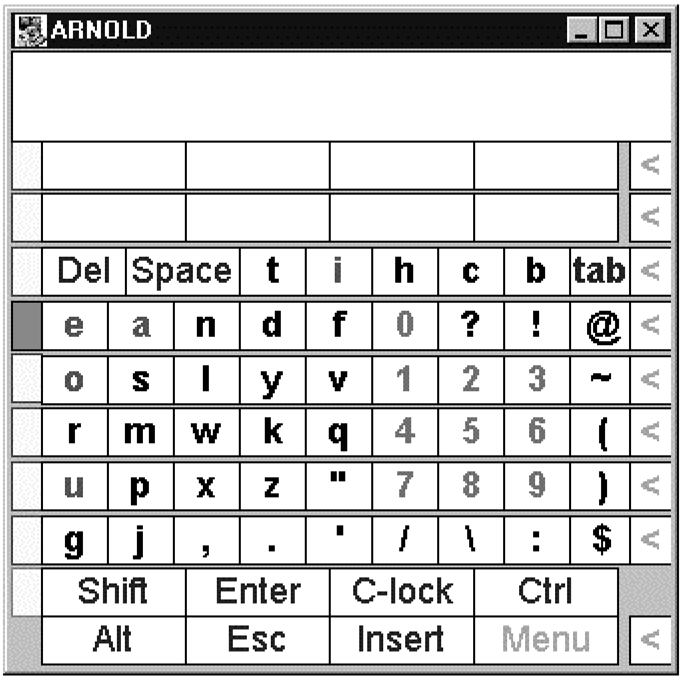

Figure 1. *Virtual Keyboard Window*

Even with our small number of users we were surprised to find significant variations in their abilities. Not only that, but there were changes in a single user throughout the day, for example through boredom or fatigue. The virtual keyboard would need to be able to be readily adaptable to these differences. Another variable is the application itself. The layout would need to be changed when moving from say a word processor to a spreadsheet. With limited funds it had to be kept to a low cost, as many commercial offerings are expensive. To this end we decided on a virtual keyboard implemented in software with a very simple electronic interface to whatever single switch the user could operate. For example, one might use a head switch while another would use a suck-blow tube. As the users could not control their eye-gaze, alternatives are needed for feedback to confirm when choices are made. Similarly, when the child is with a therapist or teacher, this helper also needs to be provided with some feedback. For example, to confirm that the switch was indeed pressed so that they realise that the child has indeed made contact and can offer appropriate encouragement. For this feedback we chose to have a light-emitting diode on the switch box as feedback to the helper. We also chose to use sound from the computer and to flash the appropriate area in the window. Finally, it showed the last input in addition to building up the complete word.

3. INPUT METHODS

The commonly used methods for input selection in augmentative assistive devices are direct selection, encoding and scanning. For a keyboard, direct selection is much the same as the ordinary method of pressing a key. Someone with a minor disability can employ a keyguard in addition to software changes to compensate for poor motor control or single-handed operation. A keyguard is a template that fits over a conventional keyboard with openings above the key tops. The user can then place their hand(s) on the template and push a finger through one of the holes to activate the key. The finger resting in the hole then avoids minor errors from tremors. Suitable software can then ignore errors such as brief key presses or repeated activations within a short time frame or allow the use of a “sticky” meta-key. For example, in one-handed operation, the shift key is first pressed and the software interprets the next key press as if the shift key was still held down. The template also assists the user in aiming at the desired key by providing protection against erroneous positioning. Although this is a useful input method for many users, it is beyond the motor skills of our group of users. They are only able to operate a single switch and a real keyboard is too complex and the key tops too small.

Another method is one of encoding. Here the user makes a sequence of actions with different sequences providing the coded input. An example of this is the equivalent of the radio transmissions using Morse Code. Morse code is produced as a sequence of short transmissions (dots), longer transmissions (dashes) and a pause to indicate intersymbol spacing. For English, the Morse Code has additionally used short sequences for the most frequently occurring letters. For the user with a disability, a similar coding scheme can be used. It is often used with a sip switch. The users have what looks like a straw. To produce an input, the user sips on the tube activating the sensor. Short and long sips make the two forms of input with pauses breaking up the sequences. This offers a number of advantages to someone with motor impairment. Once a user has learnt the code they no longer need to concentrate on the input device. It is a compact interface and does not need attention to be distracted from the task to the actual input device. It does take training before a user becomes competent. For first time users it is slow and unreliable as it totally depends on the user internalising the meaning of the sequences. Of course a visible representation of the codes can be used during the learning phase. Again, the level of control is beyond our users.

A scanning keyboard consists of an array of characters just like an ordinary keyboard. The actual device could be a real one or, as in our case, a virtual one. In a conventional keyboard any key may be selected in much the same time, i.e. it has random access. The scanning keyboard differs in both layout and method of selecting which key to activate.

Scanning is a sequential access system. That is, each key in turn has the potential to be selected. This focus moves from key to key until the required one is reached. The user then activates their switch and the character at that position is sent to the application or in some cases stored for later transmission to an application. Imagine a linear line of the 26 lowercase letters and a dwell time for each letter to allow the user to realise that this is the required letter and then a further time to activate their switch. This is clearly too slow to be acceptable. One way of speeding up the selection process is to move away from a linear array of choices, to one of grouping. Each group in turn is focussed upon, and on selecting that group the scan is then among the keys in that group. For those who can hold a switch down, some form of dual scanning is possible, though not for our particular users. Further improvements can then be made by predicting what the next most likely set of characters are and offering that always in the first group. Additionally, word prediction can be used and even the prediction of complete phrases such as "Dear Sir/Madam." This can also be applied to the use of symbols, for example the Compic (Australian) or Bliss (Canadian) set. A common way of displaying this option is as a series of rows. The system scans down the rows and on reaching the appropriate row the user activates their switch. The system then scans along the row until the desired character is reached and the user once more clicks their switch. By placing the more frequently used letters earlier in the scan sequence the overall performance can be further improved. Finally, the redundancy of natural language allows lengthy scans to be avoided if, at each selection, a set of predictions is offered based on the history.

Scanning keyboards have the advantage of being visible and use recognition rather than recall. This is less cognitive effort when compared to the systems using some form of coding. It offers those users who can operate only a single switch the simplicity of operation. The disadvantage is the waiting time for the scan to reach the desired letter. As we have also observed, it is very frustrating to have just missed your selection and have to wait for a complete repeat of the whole scan. Some users with more motor control than our users can have the means to change the direction of scan to overcome this problem. Another possibility, again requiring better motor control, is to be able to hold down a switch to mean "cycle through the groups", while a release would initiate the scan of that row. We feel that our solution, described below, offers almost as good a capability without imposing additional dexterity requirements on the users.

4. DEVELOPMENT OF THE VIRTUAL KEYBOARD

Our users were all comfortable with the concept of the scanning selection. We have used this before in some of our earlier projects at the school. Alternatives, such as direct selection or encoding, were not suitable as they require higher levels of control than most of our users have and also these other methods have longer training times.

Our particular users are unlikely to be familiar with the standard QWERTY layout for keyboards. This will not be the case for their therapists and teachers or for people who have acquired a motor control impairment later in life. However, it is the children who will be doing the virtual typing and we considered it a design advantage not to be constrained by any particular layout. For the able-bodied, alternative layouts to the conventional one of QWERTY have not shown overwhelming advantage either in ergonomics or speed. The alphabetic layout (see Figure 2), although intuitively appealing has also not shown to perform any better than the QWERTY layout. We decided that as we had a virtual keyboard it could be completely soft in the sense that it could be readily changed according to both the users and its use.

The system uses a separate template file for the layout of the keyboard. This uses a simple syntax and is easy to change with any text editor. We have provided different layouts for word processors and spreadsheets as well as an alphabetic layout in case that should be needed. Our template for spreadsheets (see Figure 3) emphasises numeric input and cell navigation. To accommodate the use of symbols such as Compic or Bliss would not be difficult, and prediction could be still be used. As you can see from the figure, the layout is not a conventional one. This will cause some minor problems with able-bodied assistants who are familiar with more conventional layouts. For our users, who have no prior experience and are never likely to need to use a standard layout, the speed advantage with a layout optimised for English more than compensates for the unusual layout and only needs a short initial learning time.

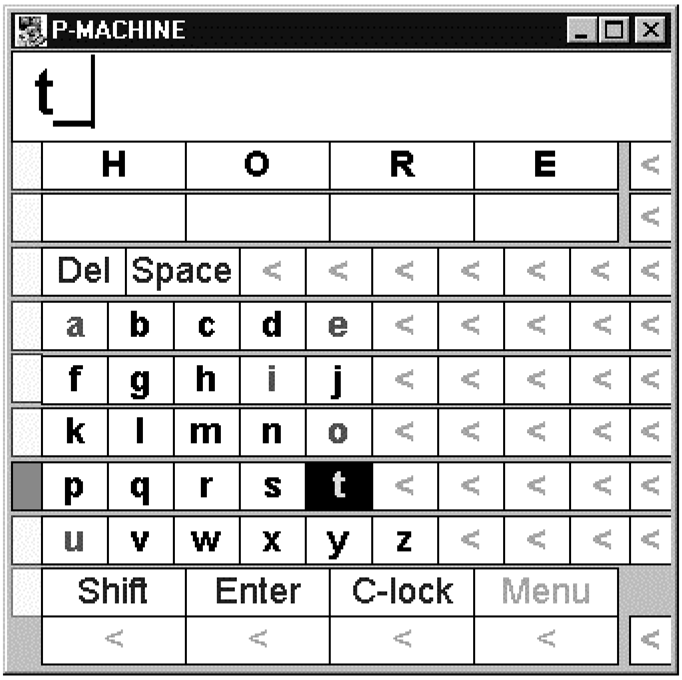

Figure 2. *Alphabetic Template Virtual Keyboard Window*

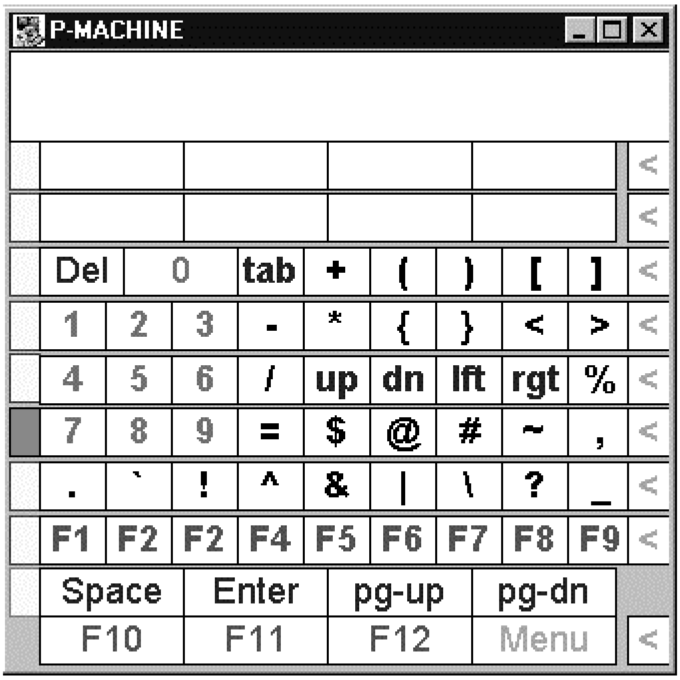

Figure 3. *Numeric Template Virtual Keyboard Window*

For the predictive keyboard we decided to provide both character and word completion. Examining the virtual keyboard (see Figure 4, with a "t" selected), the top part of the window shows (in large letters) what has been typed so far. The next row down is the character prediction row. From our analysis of English texts the system is predicting the four most likely following letters. The row below that contains four predictions to complete the current word. The left-hand column is used to highlight row selections. The layout shown is for word processing and is organised according to the expected letter frequencies found in English. Keys are colour coded to aid searching. The last row shows other keys, for example, selecting "Shift" actually changes all displayed characters to their shifted form. This is an advantage of a soft keyboard in that the key tops actually show what character will be entered when selected. We have found that the use of CAPITAL letters on conventional keyboards is very confusing for the younger child, when in fact it is only if the SHIFT key is held down do you get the capital letter. Finally, the last row has a range of keys that users may need, for instance you can send alt-sequences that can manipulate operating system functions without the need to go through ordinary menu selections.

Figure 4. *Letter "t" Chosen, with Frequency Prediction*

"Menu" allows for configuration changes (see Figure 5). Once selected the prediction rows contain the options, which may be selected in the same way as normal characters. For instance the figure shows the result of selecting "Menu". The top row is then a set of options. The first is for the frequency ordered layout suitable for word processor applications or Internet tools such as email or URL entry. The next option is to select the alphabetic layout and the final option is aimed at a spreadsheet application. The next row allows other parameters to be altered. Each child will require different settings such as speed of dwell, scanning time etc. The window itself may be resized should the user need larger letters. All the constituent parts are scaled automatically. There are tradeoffs between sharing screen space with the applications (that is, you need as small a floating window as practical) and coping with users who have poor eyesight (where a larger window would help). For younger users it is likely that a reduced set of keys is required, as few will want the full range available on normal 101-style keyboards. Limiting the predictions to only four means that visual search times are kept brief as they are constantly updated and so cannot be learnt. The fixed layout of the remainder of the keyboard is one that the user will gradually learn and spend less time visually searching.

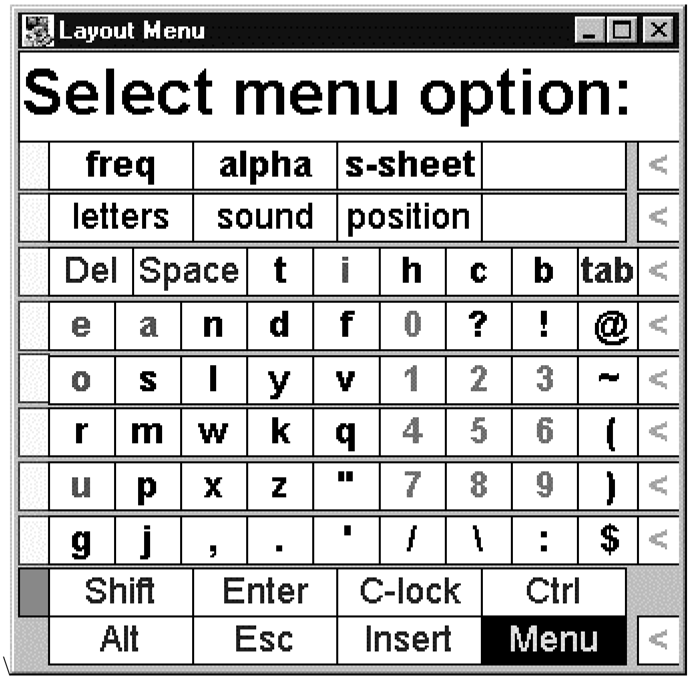

Figure 5. *Menu for Virtual Keyboard Window*

One major improvement that we have made to the scanning method is to have a “go back” symbol on the end of each scan row. A frequent cause of frustration is to miss a selection on a row, and have to wait to cycle around through all the rows again. With this simple addition the user can select the “go back” as they reach the end of the row in which they missed making the selection. The system then re-scans that row and the user has another chance at making their selection. If the row was chosen in error then the only penalty is one additional dwell time on that “go back” symbol.

Character prediction is based on n-grams. We experimented with different training texts and values for n. Performance was best with tri-grams for the English texts that we used. We found that it didn’t matter whether the text was from the user or an unrelated author. This meant that we could pre-load our frequency tables for new users without waiting for a large personal file to be built. Word prediction is possible using dictionaries with some fast form of searching. Instead, we opted for a simpler one of word completion rather than prediction, using a pre-loaded dictionary that is added to as the user types. This dictionary contains an expected frequency of use. This is updated as the user enters text. For both character and word predictions we deliberately kept the choice to four in order to avoid long search times. We found that our predictions were more likely to be correct for our users than commercial systems. For instance, if a “t” is entered the predictions are “the to that this.” One popular commercial system would offer a very long list such as “tab table tablet tabloid tabulate tachometer …”. Most of these would not be part of our users’ vocabulary and it also takes a long time to visually scan through this list to see if the prediction row contains what is wanted.

5. SYSTEM TRIALS

Our initial trials have been restricted to several able-bodied students and ourselves from our department. We wanted to evaluate the performance without confounding variables and to carry out changes without being concerned with any backward compatibility. Further, we did not want to offer our users something that was still very much an experimental prototype. We have also consulted with staff at the Cerebral Palsy Association of Western Australia who have considerable experience with a wide range of users. With the able-bodied users we found that learning was very rapid with improvements being shown on even the second trial. Errors also decreased quite rapidly due to learning of the layout and how to optimise selections from the predictions. We all appreciated the convenience of the “go back” key. Over the next few months we will be improving the prototype and will then be offering it for evaluation with a user already familiar with the type of system as he has been using a commercial system. Although our objective is to design a system for children with severe motor impairment through Cerebral Palsy, it is likely to be useable by anyone with similar levels of motor impairment.

Acknowledgements: It is a pleasure to acknowledge the efforts of Martin Masek in implementing and contributing to this project. Staff at the Sir David Brand School and the Cerebral Palsy Association were very helpful in developing the project. Finally, our thanks go to the children and our student testers in the department. The department internally funded the costs of its development.

6. REFERENCES

N. Alm, J.L. Arnott and A.F. Newell (1992), Prediction and Conversational Momentum in an Augmentative Communication System, Commn. ACM, **35**, *5*, pp. 46-56.

J.J. Darragh. and I.H. Witten (1992), *The Reactive Keyboard*, Cambridge University Press, ISBN 0-521-49375-8.

A.D.N. Edwards (1995), *Extra-Ordinary Human-Computer Interaction – Interfaces for Users with Disabilities*, Cambridge University Press, ISBN 0-521-43413-0.

S.R. Ellis (1995), Origins and Elements of Virtual Environments, in Virtual Environments and Advanced Interface Design, eds. W. Barfield & T.A. Furness III, Oxford University Press, pp. 14-57

N.A. Hine, W. Beattie, A. McKinlay and J.L. Arnott (1994), Consideration of Scanning Keyboard and Text Prediction in the Context of Access to Telecommunications Services, ISAAC: 6th Biennial Conference of the International Society for Augmentative and Alternative Communication, Maastricht, The Netherlands, 9-13 October 1994.

P.E. Jones (1997), Computer-Assisted Remote Control for the User with Motor Impairment, Proceedings of the XII BCS Conference on Human Computer Interaction '97, Springer-Verlag ISBN 3-540-76172-1, Bristol, 12-15th August, 1997 pp. 205-221.

G. Singh, S.K. Feiner and D. Thalmann (1996), Virtual Reality — Software and Technology, Commn. of ACM, **39**, *5*, pp. 35-36, May 1996.

7. URL

Origin Instruments Corporation (1998) — SofType keyboard http://www.augmentative.com/acs-styp.htm (July 1998)