Using the HeadMaster With Macintosh, Apple II, and MS-DOS Computers

Denis Anson

Key Words: computer assisted therapy • computers (use in therapy)

When it was first introduced, the Prentke Romich HeadMaster allowed a person with good head control to access a Macintosh computer. The hardware and software combination allowed the user with a disability to type, to pull down menus, and to print, but only on the Macintosh. The HeadMaster itself provided an excellent replacement for the mouse. The keyboard software, however, could be difficult to use and would not work with some programs.

In the past 2 years, new hardware and software options have made the HeadMaster a much more valuable tool for the general computer user with a disability. This article discusses these options and their strengths and weaknesses. This information may assist the therapist prescribing alternative access equipment in making appropriate hardware and software choices.

In 1984, Apple Computer introduced the Macintosh, a computer that changed the way in which people interact with computers. Instead of remembering obscure commands to make the computer do things, the Macintosh user could point to commands on menus. Use of the Macintosh involved a different approach to the user interface and a new input device called a mouse. The mouse was not a new idea—it had been used in specialty computers for some years. The Macintosh, however, was the first mass-market computer that included a mouse.

The Macintosh met with broad acceptance. The mouse, however, met with resistance. Touch typists did not like the mouse because one hand had to be taken off of the keyboard to use it, which interrupts typing. Innovators endeavored to find an acceptable alternative to the mouse.

A small company called Personics developed the HeadMaster, a replacement for the mouse that did not require the user to move his or her hands off of the keyboard. The HeadMaster device consisted of a lightweight headset connected to the computer by a wire. It allowed the user to control the position of the mouse by changing his or her head position. A small tube connected to a pressure switch allowed the user to click the mouse by puffing. Alternatively, the user could connect a switch directly to the control box and click the mouse through foot movement. A Macintosh user with a HeadMaster could use the computer all day long without moving his or her hands from the keyboard.

If computer users only had to type, the HeadMaster might have been very successful in the mass market. But office work involves additional activities, which were made difficult when one was using the HeadMaster. For example, the HeadMaster user could not talk on the telephone with the puff tube of the HeadMaster in his or her mouth. Additionally, because the HeadMaster is connected to the computer, the user had to disconnect it before walking over to a filing cabinet. Some users complained that continually donning and removing the headset messed up their hair. The HeadMaster, therefore, was not a viable replacement for the mouse in the mass market.

Although the HeadMaster was not a success with nondisabled users, it might be suitable for persons with high-level quadriplegia who wish to use a computer. Such persons cannot use the keyboard effectively, much less the mouse. If the HeadMaster can allow such persons to type as well as use the mouse, then it might be a viable access method. Robert J. Martin developed the ScreenTyper, an on-screen keyboard that allows the HeadMaster user to type as well as move the mouse. (The ScreenTyper has been discontinued and is therefore no longer available for purchase.) The combination of the ScreenTyper and HeadMaster was marketed by Personics to persons with disabilities. ScreenTyper, which placed two small rows of characters across the screen, provided access to the Macintosh for anyone who had good head control.
control. By pointing to a letter and puffing, the HeadMaster user could type that letter. Unfortunately, ScreenTyper could be difficult to use for typing because the user had to be able to point precisely to a letter in order to type it. Worse, it had a tendency to crash unexpectedly.

Since the original HeadMaster was developed, changes in the computer industry at large and in the HeadMaster system have made it a more useful device. The original HeadMaster was designed to emulate the mouse that came with the Macintosh 128, Macintosh 512, and Macintosh Plus. The mouse connector on these computers was a 9-pin D connector that provided direct access to the computer. The original mouse could be connected through the HeadMaster so that either device could be used.

When Apple introduced the Macintosh SE and the Macintosh II, it also introduced a new keyboard and mouse communications system called the Apple Desktop Bus (ADB). This new keyboard system allowed multiple input devices to be connected in a chain and to communicate with the computer in a standard way. Unfortunately, it also made the HeadMaster incompatible, because the original 9-pin mouse connector no longer existed.

To maintain compatibility, Personics introduced a new ADB adaptor—a modified Apple mouse. To use the HeadMaster with any Macintosh introduced after the Macintosh Plus, the user connected it through the ADB adaptor. The ADB adaptor can be connected to the Macintosh either through the back of the machine if the Macintosh has two ADB connectors on the back or in series with the keyboard if only one ADB connector is available. The development of the ADB adaptor signaled the beginning of several changes in the HeadMaster.

In 1989, the HeadMaster was sold to Prentke Romich. Some of the changes described below were in progress at the time of this sale. Other changes were initiated by Prentke Romich after acquiring the HeadMaster.

HeadMaster and the Macintosh

ScreenKeys

The original on-screen keyboard provided with the HeadMaster was, as noted above, somewhat difficult to use due to the small key size. In 1989, Personics replaced ScreenTyper with ScreenKeys. ScreenKeys, which is on the computer screen, looks exactly like a standard Macintosh keyboard. It has three additional features, however, that are important to the user with a disability. First, it is resizable. The on-screen keyboard can be made larger or smaller, as needed. The size of the keys can be adjusted to suit the user’s control limitations. For example, a user with good motor control can shrink the keyboard to leave most of the screen available to the application program, whereas a user with less motor control can have a larger keyboard, with keys that are easier to hit, but with proportionally less screen available for the application program.

Second, ScreenKeys, like standard keyboards, has an adjustable automatic repeat feature, so that the holding down of a key causes that letter or symbol to repeat. This is a useful feature when one wishes to use the underline character to draw lines. ScreenKeys allows the user to define how quickly a key begins to repeat and how fast it repeats once it starts. This can be easier to do than puffing once for each character, if the user has sufficient breath support.

Third, ScreenKeys allows dwell typing, called AutoKeyPress in ScreenKeys, for users who are not able to use any HeadMaster compatible switch. With AutoKeyPress activated, the user need only hold the mouse pointer on the character for a period of time for the key to be automatically typed. For example, a patient with a history of orofacial surgery prior to a brain stem stroke and no reliable switch site could type with AutoKeyPress. He or she could not, however, pull down menus, click boxes or icons, or perform any of the other basic operations that the Macintosh requires. Once the patient was set up, he or she could type. Until set up, however, he or she could do nothing.

The user activates ScreenKeys from the Apple menu by pointing to the Apple on the screen and puffing, then moving down the list of desktop accessories to the ScreenKeys entry. This opens the keyboard window but does not immediately allow the user to type. The ScreenKeys keyboard is the active window when it is first activated, so any keys pressed are sent to the keyboard. To type, the user must make the document window active. The Macintosh user can activate a window by clicking anywhere in its interior. When this is done, the ScreenKeys window seems to disappear, because it is covered up by the active window. The user must shrink the active window with the zoom box to show the ScreenKeys keyboard in the background in order to type. When this is done, the user can press keys on the on-screen keyboard and they will appear in the active window. (Note that all of these steps require the ability to operate the mouse switch.)

Unfortunately, not all Macintosh programs allow the user to shrink the document window. HyperCard, for example, uses the entire screen on the compact Macintosh. Many interesting and powerful applications are being produced with the use of HyperCard, but the user with disabilities who is typing on ScreenKeys will not have access to them. (HyperCard 2.0 does allow for the resizing of windows, but it does not allow for the scrolling of the window. It allows access only to the top half of a HyperCard stack for the ScreenKeys user.) The inability to use full-screen applications may be a major limitation, depending on the individual user’s needs. I believe that if an alternative access system does not give complete access to the computer in question, it should not be considered a finished product. Some developers are satisfied to produce products that work only with Word Perfect, dBase, and Lotus 1-2-3, because that is what they believe
persons with disabilities are using. I believe that we should provide full access and let the person with a disability decide what he or she wants to use.

ScreenKeys has one problem that will not bother most users, but might bother the serious Macintosh user. The ScreenKeys program consists of an init file, which is dropped into the system folder of the Macintosh and is activated on power up, and a desk accessory, which is activated from the Apple menu. The problem stems from the init portion of the program, so it affects all users of the adapted Macintosh, regardless of whether or not they are using ScreenKeys. On programs that use nonstandard menus, some menu items do not work correctly if the ScreenKeys init is installed. For example, many of the formatting features of Microsoft Word are activated from a dialogue box that pops up when the user selects Character from the format menu. Inside this dialogue box are secondary menus for type of underlining and line spacing, in addition to numerous check boxes for character style. None of the entries on the inner menus can affect the formatting of text in the word processor. For example, if the ScreenKeys init is installed, one cannot double underline from inside Microsoft Word. In Aldus PageMaker 4.0 and Systat 5.0, the special menus are accessible but do not work correctly. Programs that use the traditional pull-down menus work perfectly with ScreenKeys. I do not like software that introduces strange behavior into computer programs, because I find it time-consuming to try to figure out how certain functions work.

Despite these limitations, ScreenKeys is a viable program and generally works well with applications software that uses the standard Macintosh tools. For most Macintosh users, the few limitations will not pose a problem. Table 1 lists the manufacturers and prices for ScreenKeys as well as for the other products listed in this paper.

WordWriter

Macintosh users who are bothered by the limitations of ScreenKeys have an alternative: WordWriter by Macnty. In theory, WordWriter operates exactly like ScreenKeys with only a few exceptions, the most obvious being that WordWriter includes word prediction. The sides of the WordWriter keyboard contain lists of words starting with the letter sequence that the user has typed. Selection of one of these words will cause the entire word to be typed into the application window. After typing a word that is not included in the prediction dictionary, the user can add it to the list by clicking on the Add Word key.

A less obvious difference is that WordWriter is not a window—it is a palette. Although similar in concept to a window, it is never active. It is always in the foreground, on top of the active window. Windows may be thought of as individual pieces of paper scattered on a desktop. A palette is attached to something more like the sneeze guards over restaurant salad bars. This particular sneeze guard is installed over the desk, but it allows information or tools to be displayed above the desktop. Because the tools are suspended over the desktop, they do not interfere with things that are happening on the desktop.

In practice, WordWriter seems to operate much like ScreenKeys. WordWriter is selected from the Apple menu. A keyboard pops up on the screen that contains keys that the user can activate by pointing to them with the HeadMaster and activating the switch. Unlike ScreenKeys, however, keys can be selected immediately and are sent to the window that was active at the time WordWriter was activated. No amount of shuffling of windows can make the keyboard disappear under an active window, because WordWriter is mounted on the sneeze guard above the desktop.

Some programs, however, can disrupt the WordWriter keyboard, such as those that also use palettes. For example, the tear-off menus of MacPaint are actually palettes. If the user puts away the MacPaint tools by clicking the Close box, MacPaint closes all active palettes, including WordWriter. Fortunately, the user can restore WordWriter by selecting it again from the Apple menu.

Table 1

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prentke Romich Company</td>
<td>HeadMaster</td>
<td>$900</td>
</tr>
<tr>
<td>1022 Heyl Rd.</td>
<td>ADB Adaptor</td>
<td>$150</td>
</tr>
<tr>
<td>Wooster, OH 44691</td>
<td>IBM Adaptor</td>
<td>$150</td>
</tr>
<tr>
<td>216-262-1984</td>
<td>KeyWin</td>
<td>$95</td>
</tr>
<tr>
<td>1-800-642-8255</td>
<td>ScreenKeys</td>
<td>$95</td>
</tr>
<tr>
<td>Berkeley Systems, Inc.</td>
<td>ScreenKeys</td>
<td>$50</td>
</tr>
<tr>
<td>1700 Shattuck Ave.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Berkeley, CA 94709</td>
<td></td>
<td></td>
</tr>
<tr>
<td>415-540-5555</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McIntyre Computer Systems</td>
<td>WordWriter</td>
<td>$100</td>
</tr>
<tr>
<td>22809 Shagbark, Ste. 101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Birmingham, MI 48010</td>
<td>MacKeyboard</td>
<td>$295</td>
</tr>
<tr>
<td>313-645-5090</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple Programmers and</td>
<td>Video Keyboard for</td>
<td>$20</td>
</tr>
<tr>
<td>Developers Association</td>
<td>Apple HGS</td>
<td></td>
</tr>
<tr>
<td>Apple Computer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20525 Mariani</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cupertino, CA 95014</td>
<td></td>
<td></td>
</tr>
<tr>
<td>408-996-1010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsystems Software, Inc.</td>
<td>HandiKey</td>
<td>$495</td>
</tr>
<tr>
<td>600 Worcester Rd.</td>
<td>HandiKey Deluxe</td>
<td>$695</td>
</tr>
<tr>
<td>Framingham, MA 01701</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice line: 508-626-8511</td>
<td>Computer bulletin board line: 508-875-8099</td>
<td></td>
</tr>
<tr>
<td>World Communications</td>
<td>Help-U-Keyboard</td>
<td>$595</td>
</tr>
<tr>
<td>245 Tonopah Dr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fremont, CA 94539</td>
<td></td>
<td></td>
</tr>
<tr>
<td>415-656-9011</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Another example of apparent disruption of the WordWriter window occurs with some Microsoft products. (It probably will occur with other commercial applications as well, but I have not encountered it anywhere else.) Because WordWriter is a palette, not a window, the user can type while the keyboard covers part of the underlying screen. Those who like to see a great deal of the page while working might prefer to work this way. When the user types to the bottom of the screen, the screen scrolls up to show a bit more of the page. When this happens in Microsoft Word 4.0, not only does the screen scroll, but the top edge of the WordWriter keyboard is duplicated just above the keyboard. Continued typing and scrolling produces a whole series of top edges. If the text is scrolled off the top of the screen or closed and reopened, the underlying text is found to be intact, but it cannot be seen after it is covered.

This problem occurs because of a shortcut that some programs use in order to work faster. All word-processing programs maintain a buffer that holds the text that a user types. When the screen scrolls, the program is supposed to copy the current screen from the buffer so that the displayed information accurately reflects the information that is stored. Most of the screen, however, is already displayed. A clever programmer can copy most of the screen to its new location, then just copy the new portion from the program buffer, which is faster than copying the entire screen from the buffers. This usually works well and saves much time in screen updating. If part of the screen is displaying a palette, however, the shortcut will copy the palette in the correct location, and the screen now shows two top edges of the keyboard. The text in the program buffer is intact and can be redisplayed only if the user forces the program to rewrite from the buffer instead of moving parts of the screen around. Conversely, the MacWrite II display works well when WordWriter is lying over it, because its programmers have eschewed this shortcut.

Because WordWriter and the desktop exist on separate levels, WordWriter works perfectly even with programs that fill the entire screen. WordWriter allows typing into games like Type or Wheel, which occupy the entire screen on a compact Macintosh. It allows typing in HyperCard, which does not allow the screen to be resized.

WordWriter does not allow for dwell typing. To type with WordWriter, the user must be able to operate a switch while pointing to the key. The traditional HeadMaster switch is a puff switch, but it also allows other switches to be used. The user must, however, be able to use the switch without motor overflow, which would cause movement of his or her head. I have tried to use the HeadMaster with a treadmill switch and have found that I tend to move my head slightly when pressing the switch. With practice, it can be done, but it is not easy.

MacKeyboard

Another on-screen keyboard for the Macintosh is available from TASH. MacKeyboard's features lie midway between those of ScreenKeys and WordWriter. It is a palette-based keyboard that is mounted on the sneeze guard, thus it is always on top of windows. It does not have word prediction, but it does offer dwell typing for the user who cannot operate a switch. MacKeyboard is created differently from either WordWriter or ScreenKeys in that it uses only a control panel device, which affects its operation in beneficial ways.

Utilities can be added to the Macintosh in three ways. WordWriter and ScreenKeys are both desktop accessories. A desk accessory (DA) like WordWriter is installed with a program called the Font/DA Mover, which is supplied with the Macintosh system software. On the Macintosh, fonts and desk accessories are installed in a master file called the System file, which is inside the System folder of the startup disk. Desk accessories are accessed through the Apple menu within any program that uses the standard Macintosh interface. Desk accessories can be written so as to affect only themselves or to change the behavior of parts of the application that is running. ScreenKeys also uses init files to add utilities. Init files, when placed in the System folder, are loaded at the time the Macintosh is started and change some part of its behavior. The third type of utility is the Control Panel Device file (often referred to as cDev), which adds entries to the Macintosh control panel, with which the user can tailor special features of the computer's operation. MacKeyboard consists of a control panel device and a file that defines the keyboard.

Because the Macintosh is a control panel device, its installation is easy. The user drops the program icon into the System folder and drags the layouts into a convenient location outside the System folder, then reboots the computer. Installation of the MacKeyboard causes two immediate, visible changes to the Macintosh. The first is a new menu item. In every program, the menu bar now includes a menu titled MK, for MacKeyboard. This three-item menu allows the MacKeyboard to be displayed or hidden, causes the keyboard to be redrawn if it gets scrambled by another program, and will display information about MacKeyboard. The second change is a new entry in the control panel called MacKeyboard. This entry allows the user to define how the keyboard will act.

One of the interesting features of MacKeyboard is that the user can arrange for it to automatically appear when needed and disappear when no longer needed. Alternatively, it can be set up to appear in all applications. MacKeyboard is the only on-screen keyboard for the Macintosh that allows alternate keyboard arrangements. The disk comes with two alternative keyboard layouts, and although the user cannot define keyboards, other arrangements are possible. MacKeyboard
can be configured to automatically highlight the key being pointed at and to provide audio feedback when a key is pressed. The keyboard can be resized, and when this is done, the Key Caps font changes size along with the keys. The resizing of this keyboard, therefore, can assist users with visual or motoric impairments. Neither WordWriter nor ScreenKeys will change the size of the Key Caps font.

Because it is a palette, MacKeyboard can be used with HyperCard. The only problem I have identified with MacKeyboard occurs with the use of MultiFinder. If the MacKeyboard overlaps several windows, it will occasionally cause the underlying windows to change from inactive to active. One can prevent this only by making sure that MacKeyboard does not overlap other windows. An alternative offered by the manual is to use a beta version of MultiFinder that prevents the problem. (Of course, beta products do have other problems.)

MacKeyboard looks like a good alternative for persons who want to use the HeadMaster to gain access to the Macintosh.

**FreeWheel**

For completeness, I should mention the on-screen keyboard that is shipped with the FreeWheel device from Pointer Systems. This program looks and acts almost exactly like ScreenKeys. In fact, it is the same program. Both programs are produced by Berkeley Software and sold to Prentke Romich and Pointer Systems, respectively. The only difference between the two is that wherever one program says ScreenKeys, the other says FreeWheel. Otherwise, all of the features, operations, and even problems are identical.

**Making Choices**

At present, when working with the HeadMaster on the Macintosh, WordWriter is the keyboard of choice if the user can use a switch. The combination of relatively low cost, word prediction, and access to full-screen applications makes this the keyboard to use. If the user cannot use a switch and does not require access to HyperCard, then ScreenKeys should be used. ScreenKeys has a price advantage if purchased directly from Berkeley Systems and offers dwell typing, but not access to the menu bar. If HyperCard is essential to the needs of the person who cannot use a switch, then MacKeyboard will be the most functional choice. Although it costs almost six times as much as ScreenKeys, it offers access to full screen applications and dwell typing. Neither ScreenKeys nor MacKeyboard will allow the user to access the menus or to work in the Finder without the use of some switch, but both will allow text entry.

**HeadMaster and the Apple IIGS**

When Apple introduced the ADB with the Macintosh SE, it proclaimed that this would be the standard keyboard connector for all of its new computers, including the Apple II. When it introduced the Apple IIGS, it used precisely the same ADB and mouse as the Macintosh. Although the keyboard is of a different design, it is electrically identical to that on the Macintosh.

One of the major advantages of compatible connectors, in theory, is that a device that will work on one computer should work exactly the same on any computer using a compatible connector. Manufacturers, therefore, do not have to create special versions of their products for each computer they manufacture.

In early 1989, I plugged the HeadMaster ADB adapter into the keyboard on my Apple II prototype in place of the Apple mouse. It worked perfectly, but, of course, only as a mouse, because there was not an on-screen keyboard for the Apple IIGS. In one of many conversations with Bonnie Weaver, then of Personics and now a consultant for Prentke Romich, I suggested that Personics might consider developing an on-screen keyboard for the IIGS. Such a keyboard could double the interest in the HeadMaster, because many schools wanted to provide access to their Apple II computers for children with disabilities.

Bonnie told me that she had brought up that issue with the staff of Apple Computer to see if she could get information to create an on-screen keyboard for the Apple II. About a week later, she received a prototype on-screen keyboard from Apple but was constrained by a nondisclosure agreement and could not show it to anyone. The stated plan was to distribute the on-screen keyboard with the new system software.

At the Closing the Gap conference in October 1989, the staff of the Apple booth were demonstrating Video Keyboard, by Don J. Brady, on an Apple IIGS. Video Keyboard is laid out like the Apple II keyboard. It has a Close box in the upper left corner and a Zoom box in the upper right. The Zoom box on Video Keyboard does not change the size of the keyboard as such, but does fold a numeric keypad in and out on the screen. To type, the user points to the desired letter and clicks the mouse. The character is typed into the active window. Because clicking the HeadMaster is the same as clicking the mouse, Video Keyboard allows a HeadMaster to be used on the Apple IIGS.

Video Keyboard was the first application of palettes that I had ever seen. The video keyboard does not interfere with the display because it is mounted on the clear sneeze guard above the desktop. It is never the active window and cannot be covered up by documents. In at least one respect, it works even better than most of the on-screen keyboards on the Macintosh. On the Macintosh running the Finder, whenever a user moves from one application to another, all active desk accessories are...
also closed. For the user, this means that when moving from Microsoft Word to MacDraw II or from Works to Claris CAD, the keyboard vanishes and must be summoned back with the Apple menu. On the Apple IIGS, this does not happen. Once the Video Keyboard is opened and active, it stays open and active as long as the user stays in the desktop environment.

Video Keyboard is a beta product. Beta programs still have problems. Some are known at the time the product enters beta testing, and others will only be discovered after months of use. When a user purchases a beta product, it is with the tacit agreement that he or she is willing to accept the unresolved problems with the software until a final version is available. The problems with Video Keyboard are small and may even have been eliminated by now.

One of the problems is due to the nature of the Apple IIGS and its operating system. The mouse pointer has a number of possible shapes. In word processors, it generally takes one of two: the I-beam or the arrow. One uses the arrow by placing the point of the arrow on an active part of the screen and pressing the mouse button to start a process or make a selection. The I-beam is used to move the cursor to a new location when the pointer is in a text window. The shape of the mouse pointer is changed automatically when the operating system updates the screen. Whenever the cursor is moved out of a text window, it changes from an I-beam to an arrow. When it moves back into the text window, it changes back to an I-beam. When the pointer is over Video Keyboard, it is alternately in a text window and in the palette. As control of the pointer moves between the Video Keyboard and the underlying text window, the cursor flickers rapidly from arrow to I-beam. It is initially distracting, but the keyboard still works fine.

A less tolerable problem occurs when one uses Video Keyboard with AppleWorks GS. If a word processor file is opened before the Video Keyboard, opening Video Keyboard suppresses the cursor of the active document. The user cannot type in the text window until he or she closes and reopens it. As long as Video Keyboard is opened before the text window, it works fine.

Video Keyboard lacks two features that I believe it should have. It is not resizable. If a user cannot control the mouse pointer well enough to hit the keys, there is no way to make them larger. The keyboard currently covers only half the width of the screen, so it could be resizable while still allowing for usable space on the screen. Additionally, I believe there should be an option for dwell typing. Unless the user can operate a switch, he or she cannot access Video Keyboard. Of course, unless the user can operate a switch, he or she cannot use the pull-down menus and active buttons of the software, and thus will not have full access to the computer anyway.

I would like to see a way to make an on-screen keyboard operate on the text screen of the Apple II computer. Of the approximate 18,000 programs for the Apple II, more than 15,000 run in text mode. These programs, which remain unavailable to the HeadMaster user, include such mainstays as Classic AppleWorks as well as all BASIC programs from the public domain library. Unfortunately, the operating system of the Apple IIe does not support the tools that make the Video Keyboard possible. For now, the HeadMaster user has only partial access to the Apple IIGS.

At the Closing the Gap conference in 1990, Apple announced that the Video Keyboard would be made available to all users through the Alliance for Technology Access centers across the country. Anyone with access to such a site should be able to get Video Keyboard at little or no cost. Persons without such access can still get the product from the Apple Programmers and Developers Association.

**HeadMaster and MS-DOS Computers**

By mid-1989, Prentke Romich had an IBM Adaptor for the HeadMaster. It did not, however, have an effective means of providing an on-screen keyboard. The nonstandard standards of MS-DOS displays seemed to require that different interfaces be written for each program. The introduction of Windows promised to change that.

Windows is an attempt to bring a user-friendly, Macintosh-like interface to the user-tolerant environment of MS-DOS. Windows, like Macintosh, provides on-screen windows and pull-down menus. It provides a consistent means of getting information to and from the program by handling all keyboard, screen, and printer manipulation directly, so the application does not have to. Because the same program (Windows) handles the keyboard all of the time, regardless of the application program, the user should find it easy to write adaptive software addressing that one program.

**IBM Adaptor**

In 1989, we tested the IBM Adaptor (the final product will have another name) for the HeadMaster and KeyWin, an on-screen keyboard for Windows. The IBM Adaptor is an adapted serial mouse that is modified to plug into the HeadMaster. It communicates with the computer via one of the serial, or COM, ports of the computer. If a computer already uses a serial printer and a modem, it might not have a port through which one can connect the IBM adaptor. If the computer has an open COM port, the user can plug in the adapted mouse and run the provided mouse driver software to connect the HeadMaster.

On MS-DOS computers, the adaptor also requires an electrical outlet in the wall. Because power is not always available in the serial connector (although the standards say that it should be), the HeadMaster Adaptor must be
connected to an external power supply to operate. The HeadMaster Adaptor comes with a power supply that must be plugged in before running the mouse driver software. Once the HeadMaster is plugged into the mouse adaptor and the mouse adaptor is connected to the computer and to the wall, the HeadMaster works as a mouse on the MS-DOS computer.

**KeyWin**

The first version of KeyWin (a beta version) looked and operated like ScreenKeys on the Macintosh, with some interesting exceptions. The keyboard could be resized from the size of the entire screen to the size of a postage stamp. Of course, on this small keyboard, the user could barely see the keys, but they all worked. The keyboard could be arranged in a number of ways. The original personal computers (PCs) had the function keys arranged along the left side of the keyboard. The standard KeyWin arrangement follows this model. A second arrangement follows the keyboard of most AT-compatible keyboards and all PS/2 models, with the function keys across the top of keyboard. For novice typists, an alphabetical arrangement could be selected, which left the number and punctuation keys in their standard positions but rearranged the letters into an ABC pattern. The most interesting option was the modified Dvorak keyboard arrangement.

The Dvorak keyboard is supposed to minimize finger travel by placing the letters used most frequently under the fingers on the home row. A one-fingered Dvorak arrangement places the high-frequency letters in the center of the keyboard and the low-frequency letters farther away. Over time, this reduces the energy needed to type, because the typing digit need not move so far. One of the modifications made on the Modified Dvorak arrangement of KeyWin was to leave off the H key and replace it with a second N key.

The original KeyWin worked perfectly under Windows 286 and Windows 2.0. However, it did not work at all under Windows 386. The supposedly identical environments were not really identical at all. When Windows 3.0 was released, it almost worked in the most restricted mode, but still crashed.

The new KeyWin works under Windows 3.0 and has all of the keys installed. Because of the differences between Windows 3.0 and the earlier versions, users planning to purchase KeyWin should upgrade to Windows 3.0 if they have not already done so. KeyWin is the only means, to date, by which one can use the HeadMaster with Windows applications under MS-DOS, except as a mouse. KeyWin under Windows operates essentially the same as ScreenKeys under the Macintosh operating system, except that it offers alternative keyboard arrangements. The only peculiar behavior that I have noted occurs when one uses the arrow keys, and that is not the fault of KeyWin. Because not all PCs have mice, Windows has been written to allow full access from the keyboard. The arrow keys can be used to move the mouse pointer around on the screen. The effect is that if a KeyWin arrow key is pressed by placing the mouse pointer over it and clicking, the mouse pointer moves off of the arrow key. This is not a particularly useful feature, but nothing can be done about it. For the HeadMaster user, cursor movement should be performed with the mouse pointer anyway, so this is not really a problem.

To date, few MS-DOS applications run under Windows. Of the 30 or 40 word processors that run on MS-DOS computers, only 2, Word for Windows and Ami, offer full word processing in Windows. A number of programs imported from the Macintosh operate under Windows. Microsoft believes that Windows is the path of the future, so an increasing number of applications will appear that depend on the Windows environment. However, if a user really wants access to MS-DOS today, he or she needs something that will allow access to other software as well.

**HandiKey**

While I was perusing the manual for HandiKey, the scanning input package from Microsystems Software, I noted that it included a mouse input mode. Because the HeadMaster thinks it is a mouse and HeadiKey works with text screen programs, I wondered if they might work together. With a bit of reconfiguring, HandiKey and HeadMaster seem to work well for text input in standard non-Windows word processors. I contacted both HandiWare and Prentke Romich to ask if they would be interested in making the HeadMaster work with text screen software without having to do any work. Both companies were interested, and HandiKey can now be purchased with the Headmaster.

HandiKey is an on-screen keyboard that works with virtually any program operating on the text screen. A few non-Windows word processors (e.g., SamnaWrite and Word in graphics mode) actually use graphics to generate their displays, and HandiKey will not work with these. If the word processor shows italic letters in italics, it is in graphics, thus HandiKey will not work.

Like most on-screen keyboards, HandiKey displays an array, or matrix, of keys on the screen. The user points to the key desired and clicks the mouse button to select it. The key is then sent to an application. The application reads this message as if the character had come from the keyboard.

The HandiKey keyboard is a bit different from most of the on-screen keyboards in several ways. First, HandiKey is not one keyboard, but several. The keyboard characters are arranged in a set of linked matrices of related keys. One keyboard contains the letter keys, a second holds the numbers and math functions, and a third holds the punctuation and function keys. A fourth keyboard contains basic DOS functions, and a master matrix con-
tains links to all of the others. In the deluxe version, HandiKey also includes a speech matrix that allows the nonvocal user to speak any of a large number of statements through a PC-compatible speech synthesizer.

The second major difference between HandiKey and other on-screen keyboards is that it is completely user definable. Although HandiKey comes with a few basic keyboard arrangements, these are not particularly optimal for any individual. With just a bit of work, the HandiKey user can make a keyboard that exactly fits his or her needs. HandiKey is delivered with a utility called MATMOD, which allows the user to customzie the keyboard to any form desired. MATMOD allows keyboards to be designed in virtually any rectangular array, including scrolling arrays that are too large to display all at once. The individual cells in the arrays can be single characters, character sequences, or word-prediction cells. The word-prediction cells display words in an order based on the number of times that the word has been selected in the past. Character-sequence cells can contain over a page of text and keyboard commands.

Because HandiKey offers so many options to the user, it also presents a fairly imposing cognitive task. The therapist or user setting up the keyboards must plan which keys must be located on the same keyboard. He or she must learn the process of defining keyboards, although this is fairly easy to learn. Finally, the user must remember which keyboard matrix contains which functions. The perceptual task of "Where is that key on the screen?" is changed to the memory task of "Which keyboard is that character on?" If the user has the cognitive skills to learn the program, however, the program offers an incredible amount of power.

**Help-U-Keyboard**

For the MS-DOS user who does not want to learn the intricacies of HandiKey, Help-U-Keyboard offers an easier to use, though less versatile, option. Help-U-Keyboard, from World Communications, is another on-screen keyboard for MS-DOS text-based programs. Help-U-Keyboard and HandiKey offer many similar features. Both offer full access to the PC-compatible keyboard. Both offer word prediction, abbreviation expansion, and macros. HandiKey, however, offers more options, whereas Help-U-Keyboard offers more ease of use.

Help-U-Keyboard presents an on-screen representation of the PC-A T-style keyboard. The function keys are across the top of the keyboard, unlike the XT-style keyboard, which has the function keys on the left. A numeric keypad is on the right, which doubles as cursor keys. When the user begins typing, the top two rows of the keyboard are replaced by word-prediction cells. These cells attempt to predict the word the user is typing and present guesses in alphabetical order. The latest version of Help-U-Keyboard also provides next-word prediction. After the user selects a word, Help-U-Keyboard shows the words most likely to follow. The next-word prediction is based on the user's past selections.

In theory, it is possible to redefine the Help-U-Keyboard key arrangement. The key arrangement is determined by a standard ASCII (text) file. Any full-featured word processor will allow the user to modify this file. The overall size and shape of the keyboard cannot be changed, but the position of the individual keys on the keyboard can be changed through editing of the configuration file. This process, however, should not be undertaken lightly. Each cell of the keyboard is defined by the hexadecimal code for the key and for the key in combination with the Shift, the Alt, and the Control keys. If any one of these codes is incorrectly defined, it would be difficult to detect and even more difficult to fix. I recommend that the keys be left the way World Communications defines them.

**Making Choices**

No right answer exists in a choice between power and convenience. If a therapist is choosing between text screen keyboards for MS-DOS, he or she must predict the main use of the computer. If a client is going to be using the program to go to school and will be consumed with learning school material or has limited learning capacity, then ease of use will be paramount. Such a client will not want to spend the time to learn a very powerful and flexible program. Conversely, a person who will be returning to the computer at work will want all of the power and utility that the therapist can provide. For such a client, the versatility and power of HandiKey will be a good investment. Additionally, the therapist should spend some time with the client and the employer so as to learn the details of the task. This will enable the therapist to provide an initial keyboard that is efficiently tailored to meet the needs of the job tasks. For the client who will be working primarily on Windows applications, such as Excel or Word for Windows, KeyWin is the only option. The only remaining limitation for MS-DOS are programs such as SammaWrite, which use the graphics screen, but not Windows, to display text. Because each of these programs use their own techniques for screen display, it may be some time before a keyboard for such programs is produced.

**Summary and Conclusion**

In the past 2 years, the HeadMaster has evolved from a narrow to a broad access tool. The original HeadMaster allowed typing on the Apple Macintosh, up to and including the Macintosh Plus. It also worked as a Macintosh mouse. It was of limited use in schools and in the business world, however, because few schools and businesses used Macintosh. Those persons who could use it found it
to be a powerful, versatile tool, but few persons who could benefit from the HeadMaster could use it.

Today, that same HeadMaster, through third-party software and adapted mice, can be used on any Macintosh sold, with any desktop application on the Apple IIGS, and with virtually any MS-DOS application. For the client with good head control who needs access to a computer, the excellent control and low cognitive demands of the HeadMaster makes it an option that should be considered.

Home Rehabilitation Exercises

Especially for therapists working with clients to regain full range of motion...

An ideal teaching aid for clients exercising to regain full use of hands, shoulders, elbows, forearms, and wrists, these two new booklets feature clear illustrations and simple guidelines on improving strength and flexibility.

Home Rehabilitation Exercises: Hand, focuses on range of motion exercises for the hand.

Home Rehabilitation Exercises: Shoulder, Elbow, Forearm, Wrist includes range of motion exercises for the shoulder, elbow, forearm, and wrist.

Both booklets have space to note precautions based on your clients' specific needs.

$3.00 each or 10 for $25.00 AOTA member
$4.20 each or 18 for $35.00 non-member

Order now with MasterCard or VISA! Call 1-800-SAY-AOTA (AOTA members), 1-800-654-5584 (MD members), or (301) 948-9626 (non-members).