An Introduction to the Human Applications Standard Computer Interface

Part 2: Implementing the HASCI Concept

The details of an easy-to-use, consumer-quality computer console are discussed.

Chris Rutkowski
Rising Star Industries
24050 Madison St., Suite 113
Torrance, CA 90505

Will personal computers ever be as common as typewriters or automobiles? We learned last month that for the computer to evolve into a consumer product it must be both *useful* and *usable*; that is, it must be capable of improving the general quality of life and be convenient and easy to use.

The Human Applications Standard Computer Interface (HASCI) was designed to meet these requirements. Part 1 of this two-part article explained the theory and principles behind the development of the HASCI interface. This month I'll describe specific details of the interface, starting with a common feature of easy-to-use computer systems—the menu.

Menus present an exceptionally easy way of introducing the newcomer to the operation of a system. They tend to fail, however, on two points: first, some designers create unwieldy menus by trying to throw in everything but the kitchen sink; second, they provide no alternative for experienced users who eventually learn the menus cold and find it irritating to have to wait for each menu to appear.

In the HASCI scheme, the problem of cumbersome menus is eliminated by treating the entire computer system as a series of interconnected choices in an inverted tree of decisions. Each branch of the tree represents a possible function that the computer can perform for you. Also, in virtually all cases, the number of choices in a menu is kept below eight. This number of choices has proven to be a perceptual limit for understandability.

The problem of menus that make you wait is solved by allowing you to input menu selections as fast as you can make them; thus, the tedium of sitting through long, familiar menus is entirely eliminated.

The Choices

When dealing with HASCI, as with any computer system, your first choice is whether or not you want to use the computer. If you do, you must of course turn the machine on. When power is first applied, the system comes up automatically as a word processor—you needn't access the operating system.

Figure 2 illustrates the controls of the HASCI keyboard, which are divided into seven main groups of keys. Of these, the following three groups are typical of many contemporary keyboards in their configuration and layout:

- typing keys
- editing and cursor-movement keys
- the numeric keypad

The remaining four groups take the

All patents, copyrights, etc. are the property of the Rising Star Foundation, a nonprofit research corporation. For further information contact the Rising Star Foundation, 24050 Madison St., Suite 113, Torrance, CA 90505.

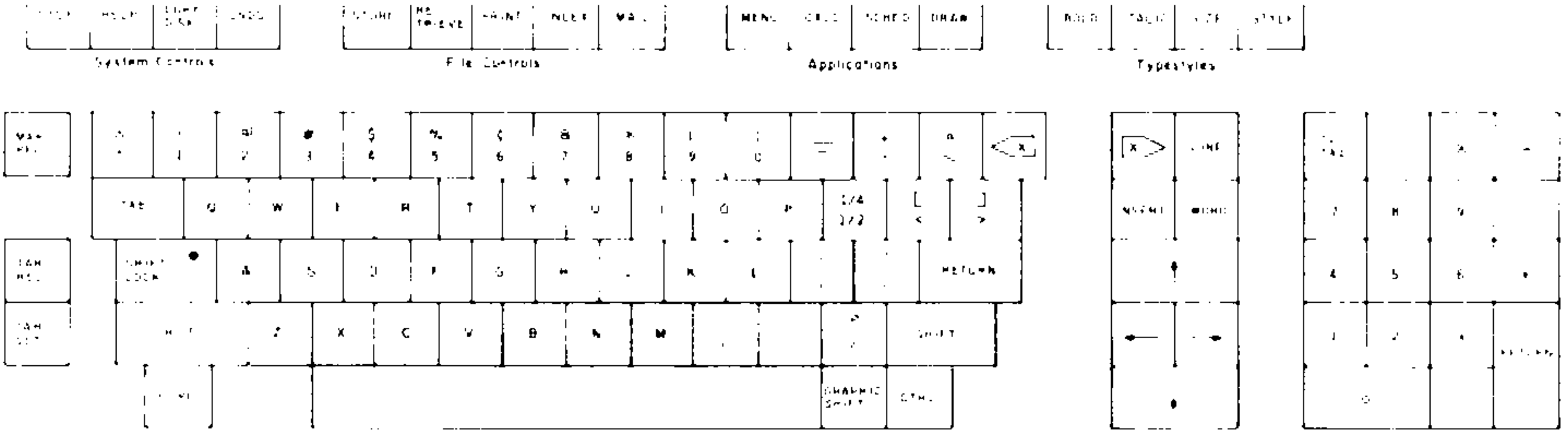


Figure 2: *The HASCI keyboard is a link between the user as intellect and the computer as tool. The interface of a computer is essentially the "handle" of the tool. All controls are clearly labeled so that the average person can correctly guess their intended functions.*

place of programmable-function keys (which have always had such clever names as F1, F2, F3, and so on). These groups give you access to the most essential functions of the system:

- system controls
- file controls
- application controls
- typestyle controls

Each of these four groups is clearly labeled on the keyboard itself. In contrast, the first three groups are self-explanatory and are not labeled.

This arrangement of the keyboard provides the first menu level of the system: you choose the *group* whose function title (or self-evident application) most closely matches your needs.

People should be able to guess which group, and which key within each group, performs any given function. The titles of the groups and the individual keys on the HASCI keyboard have been chosen to facilitate this capability. (The keyboard has been tested on a large number of people unfamiliar with computers; virtually everyone was able to correctly guess the intended function of each key the first time.) In addition, after selecting any given key, the effect on the system is immediately obvious. And, if all else fails, the HASCI system has a Help key. Thus the HASCI system is nearly manual-independent.

The second menu level involves choosing an individual key from among the seven groups on the keyboard. As mentioned before, the keys of the first three groups are already fairly familiar. Of greater interest are the individual keys of the four function control groups.

The System Controls

The four system controls affect the execution of a system program already in progress:

Stop takes the place of more usual Pause and Break keys. When pressed, it effectively halts system execution and asks if you wish to stop or continue.

Help provides you with specific information relating to the nature of the choices available at any point in the decision tree. Your options are explained in some detail. Additionally, you may access information about any specific function.

Copydisk lets you do just that: copy a disk. (We didn't use a Backup key because inexperienced computer users expected Backup to make the machine go backward.) Although Copydisk is a fundamentally necessary function in any floppy-disk-based system, this key might not be used in other implementations of HASCI.

Undo is an "undecide" key. At any point, virtually any decision can be undone with this key. It protects you from accidental deletions and also allows you to skip rapidly back up a menu tree.

The File Controls

File controls allow you to easily manipulate your files (i.e., the places where your documents are kept):

Store places a document you've created into the mass storage files.

Retrieve is the complement of Store. It allows you to procure a specific document for further symbol manipulation.

Print allows you to print the contents of any document on the system printer. Numerous print-time options are provided.

Index may be the most novel and useful key on the machine. It displays an index of all files in the system. All files are filed by date and time or sequence of creation. This information is automatically assigned by the system. The name of the file or index reference is requested by the computer in response to the Store command; you may specify a reference of up to eight words in length. When Index is pressed, you are offered three choices. You may view the index (1) sequentially by date and time of creation; (2) alphabetically by index reference; or (3) alphabetically cross-indexed, with every word in each reference cross-referenced to every other word. (This is exactly what most people wish they could do with their manual systems.)

Mail accesses a complete electronic-mail system such as the Valdocs system, which implements the HASCI system on the soon-to-be-released Epson QX-10 microcomputer. This system supports a modem and will probably support a local-area network as well.

The Applications Keys

The applications keys cover the entire family of symbol processors. If you recall from Part 1, a computer is basically a symbol manipulator, and there are four kinds of symbols that we need to manipulate: words and letters, numbers, graphic symbols, and the temporal relationships among these symbols (time).

The manipulation of words is accomplished with the typing keys and is essentially self-evident. Of the four keys in this group, three are dedicated to the remaining symbol types. These keys are labeled Calc (for calculator), Draw (for graphics utilities), and Sched (for schedule).

The nature of each of these programs is flexible: while a four-function calculator may be enough for me, you may require a sophisticated scientific processor, and a spreadsheet calculator may be ideal for someone else. Likewise, some people have simple appointment-scheduling needs while others require complicated systems such as the Performance Evaluation Review Technique (PERT) or the Critical-Path Method (CPM). The same is true for the Draw utility. Thus, no standard exists for these functions. However, HASCI standardizes the means by which one enters and departs from any symbol processor.

We didn't use a Backup key because inexperienced computer users expected it to make the machine go backward.

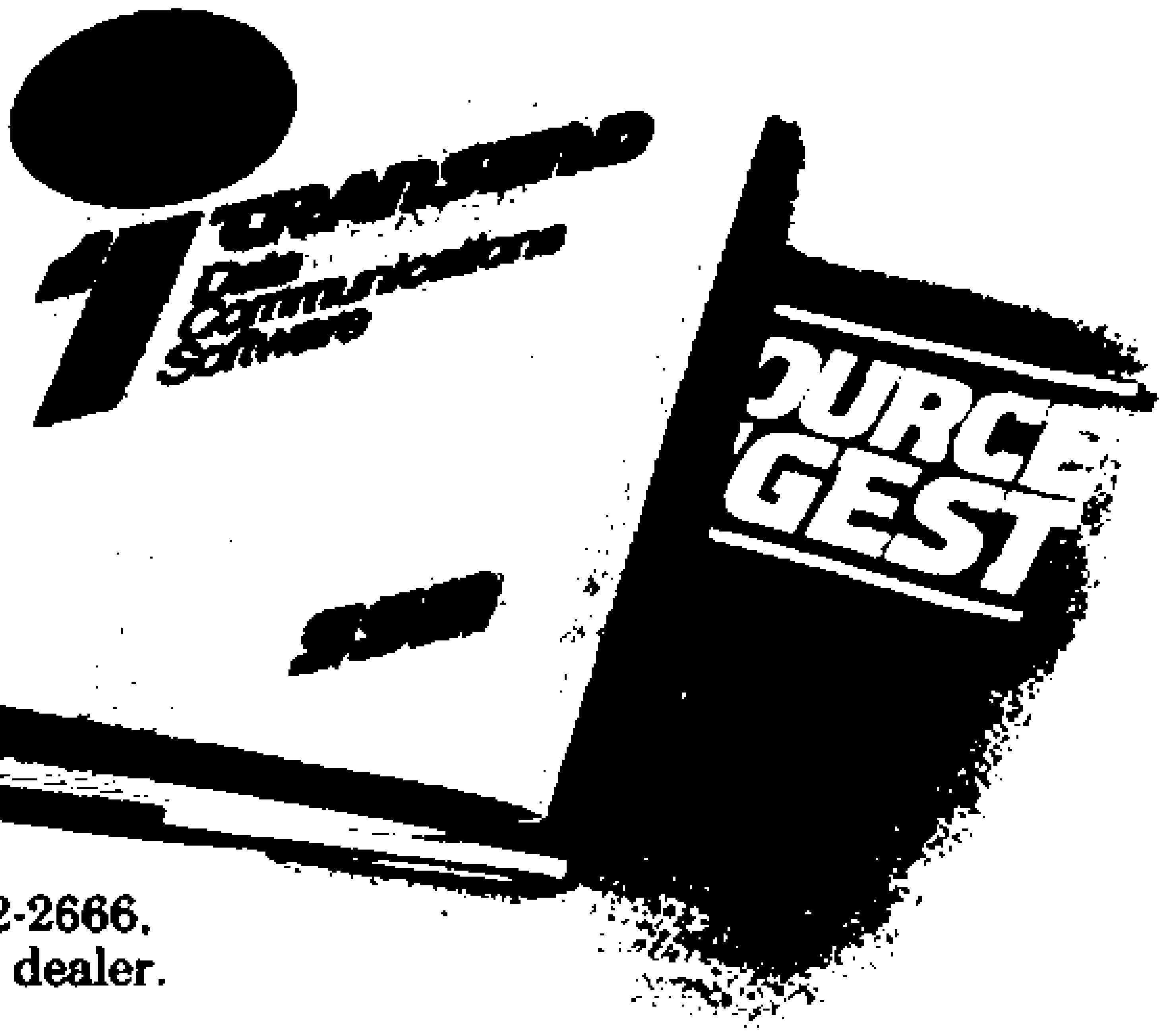
You switch from one application type to another by pressing the appropriate key. On larger systems these keys would have indicator lights to show when they were selected. The selection would also be clearly indicated on the screen.

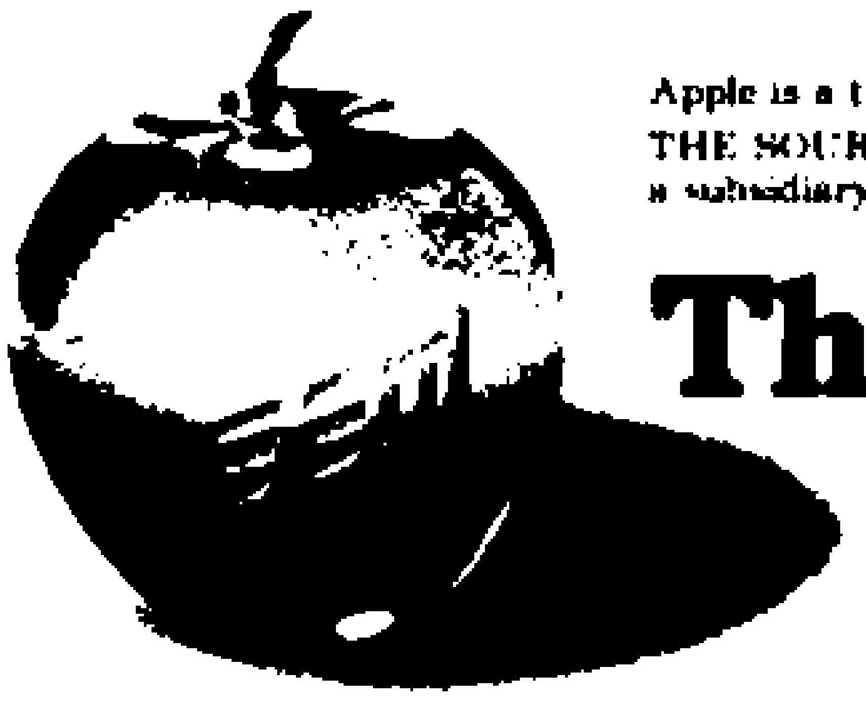
The fourth key of this group is labeled Menu. As you might guess, it is the garbage can; everything else is found there: languages, utilities, all the stuff that normally clutters up a directory listing. Ideally, any programs resident on the particular operating system that had not been converted to use the functions and protocols of HASCI would appear under Menu. In other words, a HASCI system running over CP/M should be capable of running any standard CP/M software. The same would be true for a system running Unix or any other operating system.

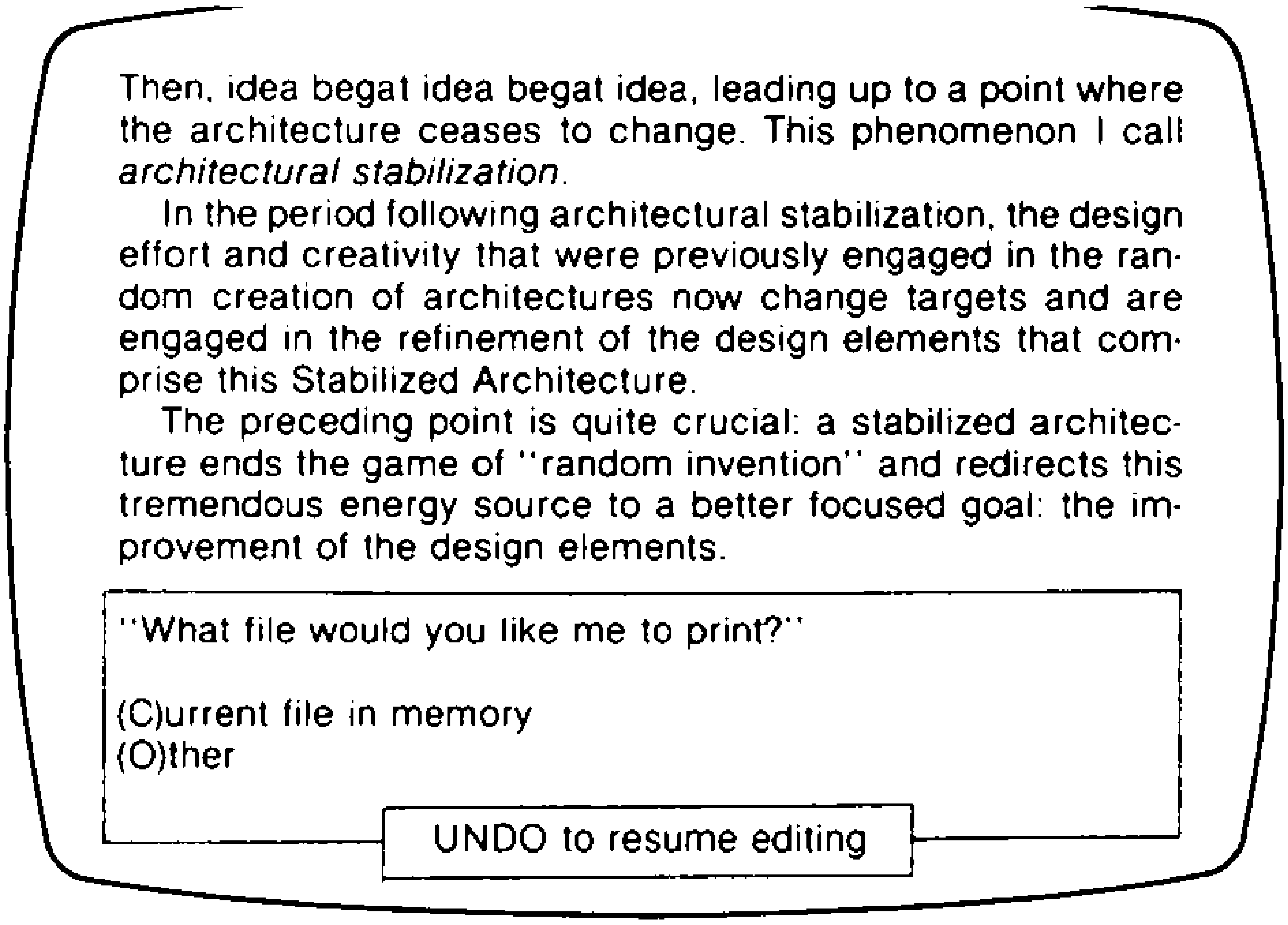
The Typestyle Keys

You can alter the symbol type style displayed on the screen in alphanumerics by using one of the four typestyle control keys: Italic, Bold, Size, and Style.

The Italic and Bold keys operate immediately. If either key is pressed







Then, idea begat idea begat idea, leading up to a point where the architecture ceases to change. This phenomenon I call *architectural stabilization*.

In the period following architectural stabilization, the design effort and creativity that were previously engaged in the random creation of architectures now change targets and are engaged in the refinement of the design elements that comprise this Stabilized Architecture.

The preceding point is quite crucial: a stabilized architecture ends the game of "random invention" and redirects this tremendous energy source to a better focused goal: the improvement of the design elements.

"What file would you like me to print?"

(C)urrent file in memory
(O)ther

UNDO to resume editing

Figure 3: *The HASCI screen is divided into three parts: (1) the document window, which contains the document being observed or manipulated, (2) the interaction window, wherein the system and the user exchange information and requests (here the system is showing the first menu after the PRINT button is pressed), and (3) the prompt window, wherein the system can place reminders about basic functions.*

while typing, all subsequent text entered will assume that type style. Pressing the key again reverts to the previous type style. The Size and Style keys access menus that allow you to select from whatever choices are supported by the terminal and printer used. In regard to Style, a machine must have at least one font; however, two (one serif and one sans-serif font) would be desirable.

The Third Menu Level

The third menu level occurs after a function has been selected by pressing its key. In some cases, there is no third level: the functions act immediately. Examples include the cursor keys and the Italic and Undo keys. Other keys may have one or more levels of menu existing beyond the keyboard. These levels are indicated on the display screen.

Screen Standardization

The screen layout should be essentially identical from menu to menu and present all necessary information in an easy-to-understand manner. The HASCI screen (see figure 3) is divided into three *windows*, each of which contains a specific type of information.

The *document window* contains the main document, which holds the symbols under inspection or manipulation. When the machine is first powered up, the display resembles that of a word processor—the document window fills the screen.

The document window may contain visual devices to simplify the manipulation of the symbol type in question. In the case of a word processor, this window contains a *ruler line*, which marks column positions, shows current column position, shows tab settings, and so forth. The window also contains a status line showing the name of the document under inspection along with more mundane items such as date and time.

When you're browsing through a file, examining a directory, or performing some similar task where you may wish to select from among many choices, the document being examined for these choices (for example, an index) would appear in the document window. If you are to make a

A Short History of the Keyboard

by Phil Lemmons
West Coast Editor

Keyboards are meant to let our fingers do the talking, but more often they make us swear aloud. Every manufacturer seems to want its keyboard to be unmistakably different from any other. The only keys that seem to be sacred and immovable are badly placed: the familiar QWERTYUIOP and its companion rows of the alphabet. The Shift and Return keys occasionally stray, and the control keys and function keys wander from one end of the keyboard to the other. Perhaps most puzzling of all, the placement of the cursor keys is not yet standardized in the most logical configuration, with the "up" key above, the "down" key below, the "left" key at the left, and the "right" key at the right. Let's hope that Chris Rutkowski's efforts to organize the placement of the most common control functions in sensible groupings will be a major step toward standardization.

If you think it's hard to adjust to a new keyboard now, though, consider the situation 80 to 90 years ago. The Gay Nineties were nightmarish for office temps. They would never know what keyboard was waiting at their next assignment. Oh, the QWERTYUIOP keyboard was around, all right, but it was only one among a hundred. Almost every company that made a typewriter used a different keyboard. Typewriters of the time (and their top-row key arrangements) included the Crandall (ZPRCHMI), the American (CJPFUBL), the Hall (KBFGNIA), the Columbia (ZKPWMCR), the Morris (XVGWSLZ), and so on, ad nauseam.

Most of the early keyboards seemed to have totally random key arrangements, but a few designs represented attempts at some sort of order. Some keyboards put T, H, and E near one another, for example, on the theory that "the" has to be typed perhaps more often than any other word. Similarly, the World typewriter keyboard put A, N, and D together, and the Edison Mimeograph typewriter grouped not only A, N, and D, and T, H, and E, but also O and F and I, N, and G.

But there's more to a keyboard than how you label the keys. Keyboards in the 1890s differed greatly in how many keys they had, how many rows of keys, and how the rows were arranged. Part of the problem was that many keyboards still lacked a shift key. Today it's hard to conceive of a typewriter without a shift key, but the idea didn't occur to anyone until Byron A. Brooks thought of it in 1875.

"Too complicated!" many people complained.

"Too tiring for the operator!" others insisted.

Such reactions prolonged the survival of some interesting mutant keyboards. The Caligraph, for example, had a circular keyboard and no shift key; all the small letters were grouped in the center and surrounded by all the capital letters, with no apparent correspondence between the arrangements of the two sets of letters. The Imperial Model B had three semicircular rows of 10 keys each, arranged in an arc convex to the typist—the central keys were nearest and the outer keys farthest from the fingertips (apparently it was designed for a typist whose outer fingers were longer than the inner ones); the layout included a shift key and a space bar, and the top row of keys read ZHJAYSCPG. Compared to the Imperial, the Hartford keyboard almost seemed to make sense: it had six straight rows of keys, no shift key, and a separate key for each small and capital letter, with identical rows of keys for both cases. The Yost keyboard, like the Imperial, had all the small letters arranged on the lower rows and all the capital letters in the same pattern in the upper rows; but the Yost had eight rows of keys instead of six. The Saturn keyboard had only one straight, very long row of keys. The Hammond had two semicircular rows, the Salter had three such rows, and the Kanzler had four gently arced rows.

One keyboard of the 1890s, the Ideal, seemed to make more sense than the others. The idea behind the Ideal keyboard was that more than 70 percent of all English words are made up of the letters DHIATENSOR; therefore, placing all these keys in one row should make typing more efficient. Some major companies adopted the Ideal keyboard, including the maker of the best typing machine of the day, the renowned Blickensderfer. (Blickensderfer's engineering prowess was such that, in 1902, it was producing an electric typewriter that used a type wheel much like the modern IBM Selectric "golf ball" or the daisy wheel.) But the Ideal keyboard, despite being as sound as a Blickensderfer, lost out to an inferior competitor, our familiar QWERTYUIOP (also known as the Universal keyboard).

By 1943, when Dr. August Dvorak proposed a clearly superior keyboard, the QWERTYUIOP keyboard had become too deeply entrenched to be easily overthrown. Dvorak's idea was to place the five vowels under the fingers of the left hand and the five most common consonants under the fingers of the right. The row of keys that resulted was AOEUIDHTNS.

Dvorak's keyboard is not the only so-called reform keyboard. The idea behind most of the reforms is to put the most common letters in easiest reach of the strongest fingers and to put the most frequenctly combined letters under the control of opposite hands. To make typing "the" faster, for example, a keyboard might put T on the right side, H on the left side, and E on the right side of the keyboard. Note that the Dvorak keyboard doesn't arrange these three letters in that way, which only proves that Dvorak wasn't trying to optimize the keyboard for typing "the."

Michael H. Adler, author of The Writing Machine (London: George Allen & Unwin, 1973), argues persuasively for a new standard keyboard that puts the 10 most common letters, ETAONIRSHD, on a single row curved in such a way that each of the 10 fingers (thumbs included) rests comfortably on one of the keys. Our thumbs now spend most of their time lolling on the space bar; Adler delegates the space bar, the shift key, *and the carriage return to the feet, freeing the thumbs for a higher destiny. This somewhat piano-like arrangement should result in much faster typing. As Adler points out, "After all. . . . a pianist can comfortably handle over 1500 to 2000 keystrokes a minute (the equivalent of 300 to 400 words per minute) on a much less compact keyboard than the one described, and without trying to break world speed records, either."*

The Battle of the Numeric Pads

Many of today's keyboards have numeric keypads—groupings of keys separate from the main alphabetic grouping—to help typists enter numbers more quickly. The numerals on the main keyboard are, of course, laid out in a single horizontal row above the QWERTYUIOP row of letters. Using the main keyboard to enter most numbers requires the use of both hands. The numberic keypad makes all the numerals available to one hand. Besides a key for each of the numerals 0 through 9, numeric keypads have a decimal-point key, a "+" key, a "-" key, and an Enter key, but for now let's consider only the numerals.

In the numeric (calculator) keypad the numerals are usually laid out something like this:

7 8 9
4 5 6
1 2 3
0

The usual keypad arrangement contrasts with the telephone company's numeric pad for entering telephone numbers:

1 2 3
4 5 6
7 8 9
0

The designers of the push-button telephone considered and tested several different arrangements of the 10 numeric keys, including two vertical rows of five buttons, two horizontal rows of five buttons, and a circle. After deciding on four rows of three keys, why didn't the designers use the traditional calculator arrangement for the numerals? Because tests established that people entered numbers more quickly and accurately with the top-to-bottom, left-to-right arrangement (perhaps because we read things in that order).

Designers of nonstandard keyboards are invited to take all these factors into account in their next designs. But a proliferation of keyboard designs would probably do more harm than good, even if most of the new designs represented an improvement on the QWERTYUIOP and calculator arrangements.

A Solution without a Standard

The programmable detached keyboard, such as those on the Victor and Epson QX-10 microcomputers, raises a new possibility: because every key on the keyboard can be programmed and the keyboard is detached, there's no reason not to have more than one keyboard for each computer. Just unplug one keyboard, plug in another, and load the operating system that loads the correct codes for the keys. This would mean that, on one computer, Harvey could type on the Dvorak keyboard with a telephone-style numeric keypad, and Eloise could use the QWERTY layout with a calculator keypad, so long as the two were content to use the system at different times. Each person could use or edit the data entered by the other; two keyboard units would be necessary only to save the trouble of relocating the key caps. This sort of flexibility would be possible on many new systems if the manufacturers would supply utility programs to enable nonprogrammers to program the keyboard. Instead of a single standard keyboard, we would have a standard of high adaptability. ■

choice from such a document, a cursor will appear to indicate that a selection is expected. But there is never more than one cursor at a time on the screen.

The *interaction window* appears only when the machine requires some discrete information or a specific response. It always appears below the document window. On an 80-column by 25-line screen, this window is 80 columns by 8 lines in size.

Two classes of interaction can occur. In the first, the computer may request a string of typed characters. For example, the system may ask, "What is your name?" The question is presented in the interaction window, along with a cursor indicating where your response will be entered. In the second class of interaction the computer requests a selection from a menu. All menus appear in the interaction window. Whenever you have to make a decision, the system prompts that explain the choices appear in the interaction window.

The *prompt window* is a small window at the bottom of the display that contains brief reminders (prompts) or flags of use for any given situation. They are optional with the software designer.

Rules for Menus

Menus must follow certain rules. First, menus should always appear in the same place on the screen. Second, menus should be designed so that you may indicate your choice by one of two standard methods: type the first letter of the first word and press the Return key, or move the cursor until it is over that letter and press Return. A third, optional method, which can be activated by a software switch, would be to type the letter without pressing Return to activate the choice immediately. The first two schemes allow the casual user simple and failsafe means of choosing from a menu, and the third method allows experienced users to reduce the number of keystrokes and access menu choices more rapidly.

Finally, menus should be organized so that the most common choices occur first in position and potentially destructive choices occur last.

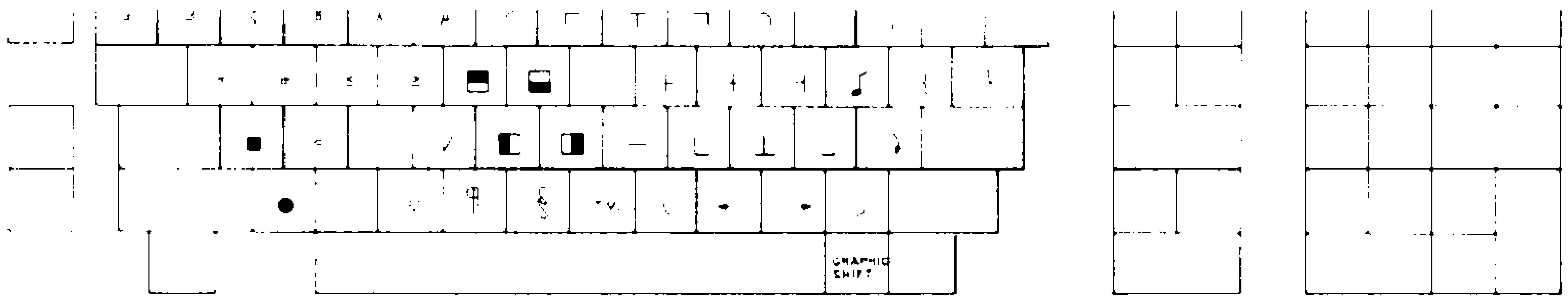


Figure 4: *The graphic symbols produced when the Graphic-Shift key is depressed. For easy reference, these symbols should be marked on the front side of the keys in an unobtrusive manner.*

The HASCI Keyboard

Like it or not, the keyboard is with us to stay. In designing a keyboard, we chose the format of the typical office typewriter. The key positions are identical and the feel is similar, so anyone familiar with a typewriter should be reasonably comfortable with the HASCI keyboard. However, by adding just a few additional keys, we were able to have the keyboard generate an entire 8-bit superset of the ASCII (American Standard Code for Information Interchange) character set. Thus the HASCI system is upward compatible with ASCII-based systems; a computer using HASCI can run any standard software.

Included in the extended ASCII is a set of standard graphic characters. These allow the creation of accented letters (by actually overtyping one character on top of another) and line drawings for boxes and forms. Also included are some Greek and special-purpose mathematical symbols. You gain access to these characters by pressing a Graphic-Shift key, which converts the normal typing keys to symbol generators. The first set of these symbols should be printed, etched, stamped, or otherwise marked on the front of the key caps in a color similar to that of the key cap. This should *not* be a high-contrast color; such treatment causes visual distraction and fatigue. Figure 4 illustrates the layout of the unshifted graphic symbols.

In addition to this primary set, one may simultaneously press Shift and Graphic Shift to access a second set of graphic characters. Most of these are logically related to their unshifted character. For example, all *line* symbols have a *double-line* counterpart. Thus, while the second set is not shown on the keycaps, it is easily learned. Figure 5 illustrates these shifted graphic symbols.

Types of Physical Controls

We have avoided using any controls other than keys and push buttons in the current HASCI standard (although voice recognition may certainly be incorporated when appropriate). Of two primary motivations the first was familiarity. Contrary to a current myth, keyboards are extremely familiar objects in our society, and a vast number of potential computer users are already familiar with their use; no other practical means of entering textual data into a computer exists today. Second, the HASCI keyboard must be available on portable computers as well as on fixed desktop units. If the interfaces

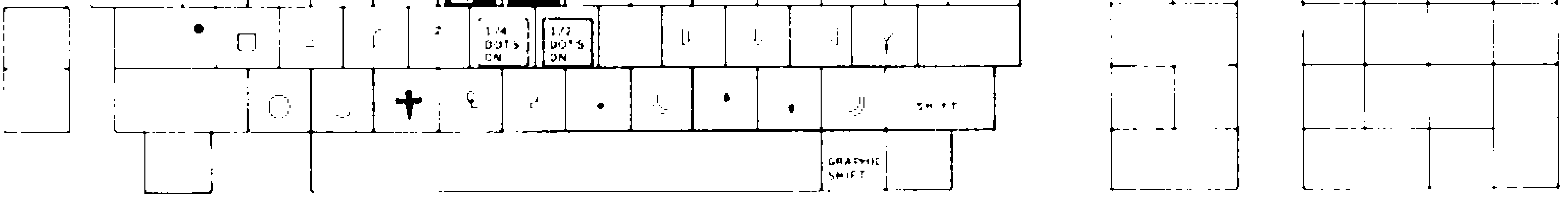
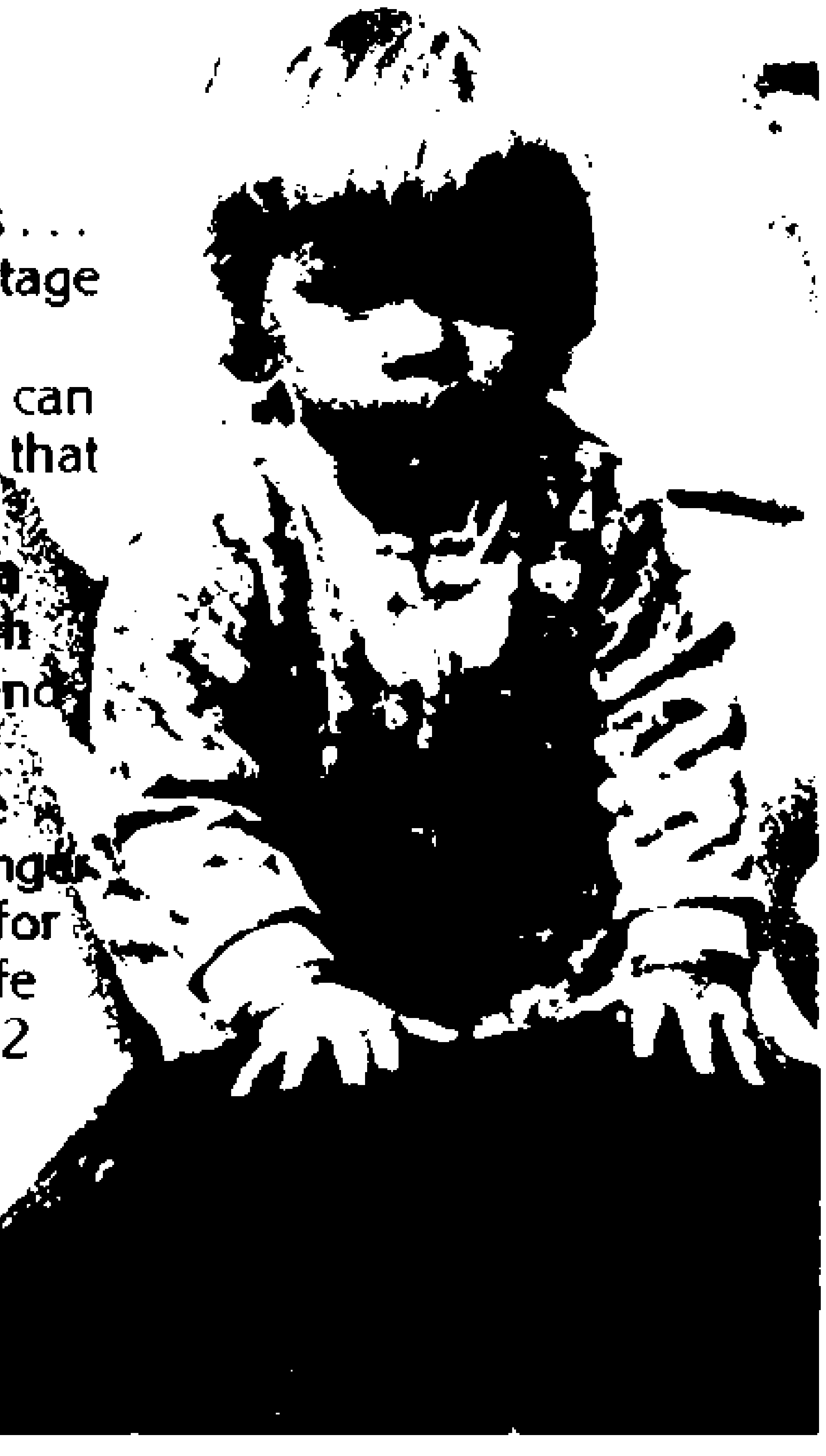


Figure 5: *The graphic symbols produced when both the Graphic-Shift and Shift keys are depressed. The symbols are not printed on the keys but are logically related to the unshifted set: single lines shift to double, fractions shift to eighth and sixteenth notes, etc. Thus the shifted set is easily remembered or learned. A certain optimal level of difficulty encourages user participation: the extremes of too easy and too difficult are both undesirable.*



on portable and fixed units are substantially different, the concept of transportable operator knowledge would be violated.

HASCI allows a number of ways of performing certain actions. No one method is suited to all possible environments. Accordingly, it is quite possible to have a Xerox-style "mouse" in a HASCI system (that's just one more way of moving a cursor around a screen and making choices). Similarly, cursor keys, control keys, joysticks, etc. are equally valid in the proper time and place. A typewriter keyboard represents merely one valid method of entering text. Others will evolve and become common, but typewriter keyboards are likely to continue in popular use for a long time.

Conclusions

The HASCI interface is by no means an end; rather, it marks the beginning of an era of consumer-oriented computers.

HASCI is not intended to be a fixed thing. We hope it will evolve and improve with time. Keys will come and go, menus will change, and groups of keys will grow and shrink. We expect that computers specifically designed from the ground up to support HASCI will help to reduce substantially the overall system cost and increase system performance.

Perhaps the best news for users is that the Epson QX-10, the first computer using HASCI, will be available from Epson America during the latter part of 1982 (see Gregg Williams's "The Epson QX-10/Valdocs System," September 1982 BYTE, page 54). And it will be very cost competitive with the current crop of personal microcomputers. ■

Acknowledgments

I would like to acknowledge the courage and support of Mr. Yasuhiro Tsubota, president of Epson America, and all the fine people of the Epson family. Without their support, the HASCI interface could not have been developed.

I would also like to acknowledge the technical assistance of Richard Mossip and Roger Amidon, whose contributions to the HASCI interface are many.