A One Key Video Graphics Terminal

Paul Frenger M.D.
A Working Hypothesis, Inc. P.O. Box 820506 Houston, TX 77282-0506

ABSTRACT:
This paper describes a complete one-button video terminal to be built into a mouse-like pointing device. A small graphics LCD is mounted on the back of the mouse. This LCD acts as a window to allow visualization of a larger virtual display surface implemented in software. Moving the terminal along the surface displays the area of current interest within the window. A variety of selector techniques may be employed, including: graphic icons, pull-down lists, pop-up menus and so on. Only one button is required to highlight and/or select the desired choice shown on the display. The device may be used as a terminal in a larger computer system or as the complete user interface in standalone handheld instruments and calculators.

INTRODUCTION

During the course of a recent[1] GEnie Forth Roundtable, Chuck Moore discussed a three button keyboard which he suggested would be sufficient for most application interfaces. After all, he had been using a seven button keypad for years prior to this. I was intrigued by this concept and immediately set about to pursue this reductionist approach to its logical conclusion.

A "zero button" keyboard is physically possible, but not intuitive (I call this the DD/SS or "dynamic deselect, static select" principle, which should be a topic for another paper). Two or more buttons suggests redundancy, so one button must be the practical minimum. However, a single key standing all alone is not as useful as a one button "terminal", complete with display. This concept will be illustrated more fully below.

BACKGROUND

In 1976 I attended the NCC (National Computer Conference) held in Dallas, Texas. The Xerox Star Computer System was shown there. Later systems based on the Star design, such as the Apple Macintosh and MicroSoft Windows, with their mouse-based graphical user interface (abbreviated G.U.I., pronounced "gooey") are convincing examples of the utility of the graphics hand/eye approach to user interface design.

A company called Industrial Electronic Engineers recently began marketing the Pixie Graphics LCD Switch, a pushbutton into which is built a 24 by 36 pixel graphics display. It is intended for use mainly in industrial applications such as control panels. My original contribution is to mount a graphics LCD onto the back of a one button mouse, to create a new entity: the one key video graphics computer terminal. The Pixie is not difficult to interface[2], but its resolution is limited. More pixels are needed.

PROPOSED IMPLEMENTATION

The current terminal design is shown in Figure 1. The mouse pointing device is the (movable) chassis of the terminal, connected by a wire cord to the computer system when not used in stand-alone mode. On this foundation is placed a high resolution

graphics LCD, mounted in a pivoted case. The sole button or key is placed conveniently at the front, to be actuated by the user's thumb as he or she views the screen in the space between thumb and index finger.

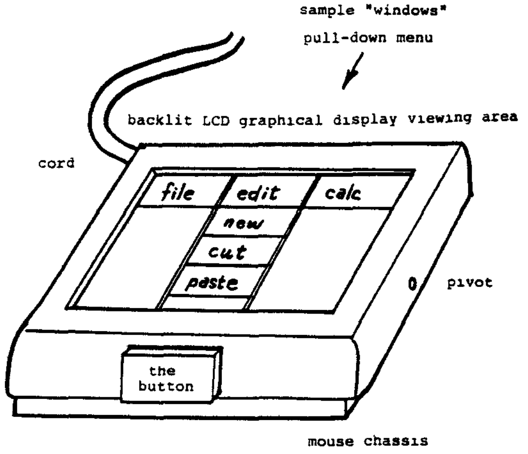

Figure 1
One Button Terminal

In practice, the user would grasp the device with either hand, rolling it along the desktop in the X-Y plane. The pivoting case could be tilted for easy viewing. A rheostat (not shown) would adjust the contrast of the wide angle (options: supertwist, backlit, color) LCD. Figure 2 represents a side view of the terminal.

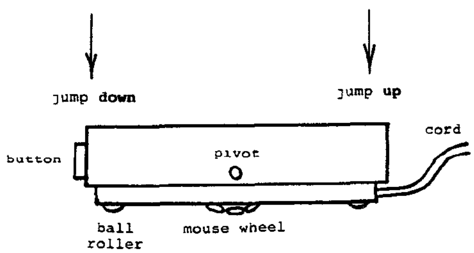

Figure 2
Side View

Figure 3 shows how movement of the terminal in the X-Y plane allows sequential viewing of a much larger "virtual display area" by scrolling.

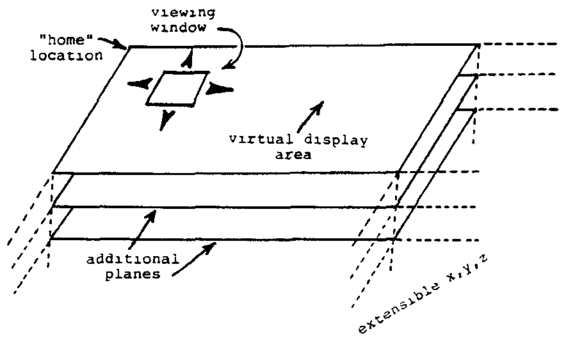

Figure 3
Virtual Display Planes

Since the virtual display is under software control, it is extensible. There is a "home" position the window assumes when initially powered up or reset; I have shown it in the figure at the upper left corner (limit) of the virtual display area.

Figure 3 also shows the presence of additional virtual display planes below the top plane. These might be accessible in a variety of ways. One method was implied in Figure 2: the pivoted display case may be rocked forwards to "jump down" one plane, or rocked backwards to "jump up". This is reminiscent of the "leap" keys present on the Canon CAT computer (which is programmed in Forth).

One-button purists might complain about these rock and roll tactics being a way to add "Orwellian" switches covertly. For this reason I suggest another technique for moving between display planes: the Escape Sequence. A pattern of button tapping (ie: three quick taps) would engage an Escape Menu, to include choices discussed below. The Escape Sequence Menu would be triggered or trapped by careless attempts at double-clicking the mouse button; choice #0 recovers harmlessly from this error. Choice

Escape Menu

0. Resume current operation.
1. Help me.
2. Jump up.
3. Jump down.
4. Reset / restart.
5. Cease current operation.

#1 moves to the HELP display plane. Choices #2 and #3 satisfy the anti-rock contingent. Choice #4 is a "HOME WINDOW" command. Choice #5 exits the current mode altogether (if allowed).

You will notice that in Figure 1 the graphics window contains a Mac-like pull down menu in it (reverse video highlighting not shown for clarity). Other selection techniques are readily implemented. Reference #2 and the IEE sales literature show the use of icons for choosing items. Icons can be zoomed from small to large and vice versa for convenience. They also can be modified dynamically (a good example is the Macintosh "ticking watch").

Even simpler menuing may be employed: ordinary text lists such as the Escape Menu mentioned above, or PC-style pop-up windows can be created. These can be in the form of dynamically alterable menus which save time and effort to find / select. a "hypertext / hypermenu" effect is possible.

PRACTICAL CONSIDERATIONS

Although the "Pixie + mouse" concept is easy to build, it's much less capable than what we want to achieve. How can the full-function version be built? This section will explore some possibilities.

First, note that video dots require the support of memory bits; this seems inescapable. A non-trivial amount of ROM and RAM memory must be present onboard this terminal. A Pixie switch has 864 pixels for its 24 by 36 dots of resolution; our hi-res display should be closer to 4000 as a practical minimum of 80 by 50 dots (ie: one quarter of an IBM PC's CGA display). Doubling this resolution in the same sized window requires four times the memory bits. Extra Z-planes and / or color further increases memory needs. This is just for the viewable portion of the virtual display areas; the total pixel support will probably be much greater still.

Compression techniques can be employed to save memory bits; the tradeoff involved is the extra time needed to decompress these figures for display. A PostScript-like description language would be a place to start (Adobe's PostScript is a Forth-like language in many ways).

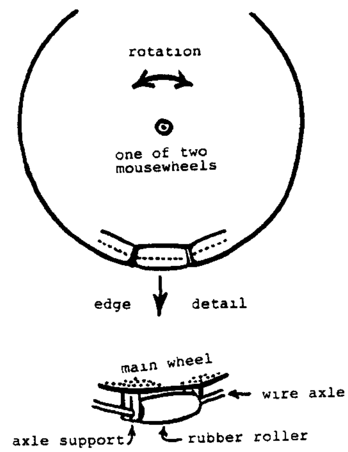

Figure 4
Mousewheel Detail

Field-upgradable memory would be helpful; for example, the credit-card sized semiconductor memory cards currently available from 8 kbytes to 512 kbytes[3]. I have drawn the mouse's positioning sensors as novel

wheels instead of trackballs in Figure 4 because two wheels forming right angles at the edges of the mouse chassis use up much less interior space than one trackball unit. This leaves room for a memory card to be slid into the chassis below the LCD control circuitry. Each wheel senses movement in a single plane of rotation, hence they must be positioned at 90 degrees (ie: North and West edges of the chassis).

Integrated circuit controllers for LCD's are commercially available (see Supplier's List). The CY325 can display 128 by 240 pixels; the W7001f can control 720 by 512 pixels. Custom devices can be created to manage the display as is currently done with graphics calculators.

THE FORTH CONNECTION

Since an onboard CPU will be required anyway, why not integrate it into the graphics control system? For example the Motorola 68000 processor has been employed in an electroluminescent touch panel[4]. Interestingly, the device has Forth built into it. A 24 bit address capability guarantees that the 68000 can control a large display area.

Refer for a moment to the GEnie Forth Roundtable mentioned earlier. In it Chuck Moore alluded to his new 32-bit CMOS Forth engine, code named "Sh-Boom". He spoke of a 50 mhz clock speed, $20 each CPU with a 1 mbyte DRAM direct interface. When this product becomes available, it will be a natural choice for this terminal project. With 20 MIPS and a large address space it can act as local CPU and LCD controller simultaneously. Other Forth engines might also work, but the cost / benefit ratio of the Sh-Boom is just right.

We have already mentioned the Forth - PostScript similarity. The one key video terminal can be programmed in any language, to support applications in any language. My preference, for a variety of reasons, is Forth on Forth. Looking at it another way, the terminal would create a "tethered" Forth system when coupled to another computer, capable of receiving downloadable fonts or display maps, programs or data, and uploading input events, X-Y coordinates, etc.

DISCUSSION

Visit your local Radio Shack store and look at their portable color LCD TV sets. Then imagine that you are peering into the viewing window of your one key video graphics terminal. Move the 3 inch screen TV along the counter while further imagining that a macintosh-like "WIMP" (windows, icons, mouse, pull-down) menu is being shown. Click any button you find handy. There, now you see how powerful the concept can be.

Not only is the main idea behind this project practical, but there are many extensions which can be added to it. Things like a "hunt and peck" virtual ASCII keyboard available in one display plane, possibly actuated by a touch sensitive display window. Something roughly similar already exists for the Macintosh[5], and a new CD-ROM video playback unit which is IBM-AT compatible[6].

How about a cordless version, like the infrared "Manager Mouse Cordless" from Numonics for PC's? How about standalone versions for use as calculators and as various kinds of scientific instruments (where the graphics display could be used to plot waveforms, FFT's, etc)? How about versions with add-on accessories (modems, voice I/O)? What about medical equipment based on such a sophisticated chassis? There are a wide variety of permutations on this theme which can be developed.

In conclusion, this new terminal concept (which is currently Patent Pending) can be a powerful method for interfacing the user to

his or her applications. Hopefully, this paper will serve as a springboard for discussion inside and outside the Forth community.

ACKNOWLEDGMENT

I would like to thank Rick Hoselton, ACM SIGForth Secretary/ Treasurer, for his valuable encouragement, comments and insights, without which I would possibly have failed to recognize the significance of this novel idea.

REFERENCES

1. "Forth Roundtable", General Electric Network (GEnie), Charles Moore, Aug 24 1989.

2. "Design a Minimum Space Auto-Instrument Module", Ralph Ong, Electronic Design, Vol.37 No.18, Aug 24 1989, pp.53-60.

3. "Semiconductor Memory Cards for Forth Systems", Paul Frenger, Proc 8th Ann Rochester Forth Conf, June 14-18 1988, pp.60-61.

4. "EL Flat Panel Subsystem Embeds Forth Kernel", EDN News, Vol.34 No.17A, pp.36.

5. "Compact Mouse Replacement Can Execute Macro Commands", Product Design & Development, Vol.44 No.10, Oct 1989, pp.66.

6. "CD + AT", Computer News for Physicians, Nov 15 1989, pp.C-4.

SUPPLIERS

Pixie Graphic LCD Switch
Industrial Electronic Engineers
Component Products Division
7740 Lemona Ave.
Van Nuys, CA 91409-9243 USA

Mac UnMouse
Microtouch Systems, Inc.
55 Jonspin Road
Wilmington, MA 01887

Manager Mouse (cordless)
Numonics, Inc.
101 Commerce Drive
Montgomeryville, PA 18936

CY325 LCD Windows Controller
Cybernetics Micro Systems, Inc.
P.O. Box 3000
San Gregorio, CA 94070

Wacom W7001f LCDP Controller
Wacom ASIC Center
4-1-12 Ikebukuro, Toshima-ku
Tokyo, 171 Japan