

STARS

3-D

in Action!

16K Disk

by Donald E. Glover

I was looking for something to do with my shiny new Action! cartridge when I ran across the article **Stars 3-D** by Craig Patchett in *ANALOG* No. 16. To become familiar with the new language, I decided to translate this demonstration program into Action!, a job I thought would take one or two hours. The task eventually took much longer, due to a number of strange quirks associated with the Action! language. I hope this discussion of my problems will save other Action! programmers some hair pulling and nail chewing.

My first task was to find a place for the display list (DLIST) and screen memory (STRLIN). I wanted to put them in a safe location, while allowing easy access from Action!-generated code and in-line machine code. I finally decided to put them in Action! arrays whose starting addresses were defined such that the display list and screen memory started on 1K boundaries in high memory. (The Atari cannot easily deal with a display list which crosses a 1K boundary or screen memory which crosses a 4K boundary.)

Calculations to generate the display list required that the address of screen memory be divided by 256 to obtain the high byte of the address. Performing this division on addresses greater than 32767, unfortunately, gives the wrong answer, since Action! multiplications and divisions always assume they are acting on signed numbers. Try typing:

```
X PRINTCE(32768/256)
```

in the Action! monitor and see what you get. After

figuring out the problem, I replaced the division by 256 with "RSH 8" (shift cardinal number right 8 bits). The use of this technique can be seen by examining the procedures STRINI() and DLSINI().

The next problem was to insert the addresses of the arrays STRTPH, STRTPL, and STRPOS into the machine language procedure SCROLL(). My initial attempt to do this involved inserting the address during the compilation phase. Using this method, the first instruction in the procedure SCROLL() would be:

```
$BO STRTPL ;LDA STRTPL, X
```

To my horror, the addresses of arrays compiled into the code by this technique frequently (but not always) differed from those observed after compilation. Apparently, the addresses of arrays change during the compile phase, and the compiler cannot modify addresses inserted into machine code. The solution was to "POKEC" the addresses into the machine language routines during run time [see the procedure MAIN()].

I believe everything else in the listing is understandable, because I kept the names of all routines and most of the comments the same as those in the original assembly language listing. A word of warning: this program is designed to work with a machine having 48K of memory. If your machine has less memory, you will have to change the starting address of the arrays DLIST and STRLIN. The place to do this is clearly marked in the listing.

Before finishing, I should mention another couple of Action! peculiarities.

