routine, and a new target will be on when exiting. This routine in turn calls the next subroutine, at locations 03D4-03E4, which cancels the wrap-around effect of the target. This effect made the movement of the target difficult to control, and detracted from the realism of the game. If you want to see whether or not you agree, enter an 00EE at location 03D4 (Return from subroutine instruction). You can also try a version which eliminates the calls to this routine in the Display New Target Subroutine. Or perhaps you'll want to include a feature in which the player can select wrap-around or not. What would be a good way to permit this? Do I hear someone suggesting self-modifying programs??? Good Heavens!

The target destruct sequence at 03E6 blows the target up after it has been hit. Each pattern of destruction was first drawn and then stored in the data section (beginning in 05C6). The patterns are displayed in order by repeated calls to the next subroutine, simply changing the "I" pointer for each pattern. The Display Debris subroutine at 0418 also calls a timer subroutine (described later in the program details given for Surround). Thus we have subroutine nesting here to a depth of three.

The section at 0406 randomly positions a new target in one of the four corners of the display, and at 0422 the Target Returns Fire subroutine begins. This routine, which lasts for less than half a second, produces an effect I really like. When you're running the program, look closely at the phaser fire coming from the target. Notice the transparency. This 3-D effect is easily achieved and adds realism to the graphics. The target's phaser is really a series of three blocks, each larger than the last. These are displayed in a cycle of four, and each block is erased immediately after it is displayed. This gives the appearance of the phaser coming toward you (any object appears to increase in size as it grows closer). To give the illusion of transparency, the target is erased