at any time, since it doesn't provide any data to the rest of the program. You could, for example, decide to write a new destruct routine, with a more elaborate explosion sequence which throws all sorts of garbage out into deep space each time you score a hit. And you simply replace the subroutine call with a call to your own subroutine to complete the "interfacing" between this existing program and your enhancement.

To continue: At location 0374, you will notice a flag - V1 - being set equal to 1. This was previously set to zero, and provides a signal to the phaser firing routine that a target was hit and no longer exists on the screen. As the target coordinates are updated between each of the phaser "bips", the routine must know if there is a target on the screen or not. The target continues to move while firing. Then, when the phaser fire routine is repeated by the subroutine call at 0376 in order to erase the phaser lines, the sub will move the target around or not, as needed.

Lines 0378-037E use a technique described in issue 1, Volume 1 of The VIPER, on making random decisions as to program flow. For your information, I have included the percentages to show the frequency with which the target will "decide" to return fire. This frequency may be changed with a new "KK" mask at 0378, but I don't know why you'd want to have it firing more often. It hits me all the time!

Next, either the target is erased or the instruction ignored (depending on the value of V1). End-game conditions are checked, and if the conditions are not met, the game continues with a jump back to 0332. If you feel your starship is being destroyed too quickly, change the value of the skip condition at 0384 to a higher number. Or live dangerously and choose a lower value.

The Fire Phasers Subroutine at 038E is a very simple display of bips angled toward center screen. The phaser length can not be easily changed, as the two sides must not run into