

different CXKK masks, but a little change here will produce drastic effects. An easier way to change the competitiveness of the game is to adjust the number of times the starship gets hit before it is destroyed.

The technique of nesting random decision points may be used to weight a complex decision "tree" in which many decisions are affected by the decisions made elsewhere. This is helpful in game simulation type programs. An excellent game of this type is "The Oregon Trail", which appeared in an issue of Creative Computing. The game is based on actual diaries and recorded frequencies of events, and demonstrates how such real situations can be translated into a decision tree. You need a BASIC-speaking computer to run that particular game, but the same concept can be applied to your VIP programs.

The remaining two CHIP-8 subroutines simply display the score and provide a timer for graphics and for the message routines.

At 054E, we wait for an instruction to start a new game - by pressing Key F. The display is simply erased and a jump is performed - back to the beginning at 0300. Please note the new erase instruction, 0230, which replaces the 00E0 instruction used for one-page CHIP-8 programs.

The only machine language subroutine in Space Wars is used to flash the screen on and off to show when the starship has been hit. It was written at location 0558 in machine language strictly for speed, and simply flips the display pages back and forth to an area completely filled with hexadecimal FF's. The reason for the two different routines - one for 3K and one for 4K systems - is one of space. In a 3K system, there isn't enough room to store two pages of ones. Playing a hunch, I tried switching to non-existent RAM locations directly above the display. The VIP sees all logic 1's at this point (I assume it's because of ground connections), and we get two pages of white that aren't really there!