

MOON PATROL / APPLE II/II+/IIe/IIc

Memory Map of game data

CODE (approximate locations)	
\$00-\$FF Zero Page	Variables for game defined in "moon" source file
\$0800-\$0E9F (CODE)	OBJ1 - produced by source file "moon patrol"
\$0EA0-\$1751 (CODE)	OBJ2 - produced by source file "moon patrol2"
\$1752-\$232E (CODE)	OBJ3 - produced by source file "moon patrol3"
\$232F-\$2F35 (CODE)	OBJ4 - produced by source file "moon patrol4"
\$2F36-\$3B64 (CODE)	OBJ5 - produced by source file "moon patrol5"
\$3D00-\$3DFF (CODE not represented by source)	Picswap binary hand coded on game disk This moves the option screen into HGR2
\$3F00-\$3FFF	Mountain data. This is loaded on game disk, there is no source. It is X/Y positions of a continue line of the blue mountains coordinates a few screen widths wide.
\$4000-\$5FFF (HGR2 where game is shown)	HGR2 - this is where the options screen and the main game play out.
\$6000-\$64FF	Tables
\$6000-\$60FF	Lookup table of shift positions to calculate what pre-shifted shape to put in HGR2 based on Xpos
\$6100-\$61FF	Lookup table of byte positions to calculate what byte offset to place shape on screen based on Xpos
\$6200-\$62FF	Lookup table of shifted shape address info blocks "INFOBLOCK" - those start at \$6C00 and are \$10 long
\$6300-\$63FF	LOWADR - low 8 bits of HGR2 lines addresses in order top to bottom
\$6400-\$64FF	HGHADR - high 8 bits of HGR2 lines addresses in order top to bottom. Combine high and low to get address of the horizontal line starting at y=0
\$6500-\$6BFF	Shapes that don't need much horizontal (X) movement, so they are not shifted and have a width/height/data format. Was needed to conserve space and speed up the game.
\$6C00-\$7FFF \$77FD (point to by \$6c04) This area is where the SHIFT1-6 data starts and continues until all shapes (that are horizontal, fast moving types) are shifted. Essentially SHIFTO exists when all modules are loaded, and in the first 30 lines of code, Moon Patrol calls "shiftshapes" to pre-shift SHIFTO to create SHIFT1-6. This allows very fast drawing irrelevant of X-position on the screen. Because HIRES pixels are represented as single bit (white) or two bit (orange/blue - or purple green), shifting on the fly is expensive unless you have a large table lookup (which in future games with more space that was done).	<p>INFOBLOCKS of all pre-shifted shapes. \$10 long The game will shift all the shapes in memory at start. This area contains \$10 long blocks of information in the following format.</p> <p>BYTE0 - Width BYTE1 - Height BYTE2 - LOW ADDRESS OF SHIFTO (base shape) BYTE3- HIGH ADDRESS OF SHIFTO BYTE4/5 - Address of SHIFT1 BYTE6/7 - Address of SHIFT2 BYTE8/9- Address of SHIFT3 BYTE10/11- Address of SHIFT4 BYTE12/13- Address of SHIFT5 BYTE 14/15- Address of SHIFTO</p> <p>NOTE: at \$6c00, the INFOBLOCK there is special because it contains two extra bytes at \$6C04 and \$6C05. Those point to \$77FD, which represents the memory space to start putting shifted shapes when moon patrol pre-shifts SHIFTO based shape to create SHIFT1-6.</p>
\$8000-\$83FF (note data actually stops at \$8347)	LAND Data - graphics data for the 3 lines of scrolling land. This data is loaded at \$8000 on the game disk, and moved by the first 20

	lines of code to \$BC00-BCFF where it resides for program execution
\$8500-\$9150	<p>This is game data, produced by source file "moon patrol 6". Course data, "Moon Patrol" compressed graphic for demo mode, demo key play table for keyboard simulation in demo mode, music notes for songs and travel music, etc.</p> <p>The data is moved to \$B000-\$BC00 during the first 20 lines of code execution in source file "moon patrol" where it resides during game execution.</p>