```
0001                        *  Move Block .02
0002                        *  By Adam Trionfo, based on a program by Eric Beckett.
0003                        *  July 20, 2010
0004                        *
0005                        *  This is a re-written version of the Move Square program by Eric
0006                        *  Beckett from the Aug, Sept, and Nov, 1984 issues of the APF
0007                        *  Imagination  Machine Newletter, "I-M 1 in a Million."  That
0008                        *  version required BASIC.  This version runs as a cartridge.
0009                        *
0010                        *  This program is written to be run on the circa 1978 APF
0011                        *  MP-1000 computer game console which uses the
0012                        *  Motorola 6800 8-Bit CPU.
0013                        *
0014                        *  Program Description
0015                        *  -------------------
0016                        *
0017                        *  The APF Menu gives one choice to the user: 1) RUN PROGRAM
0018                        *
0019                        *  This program places a white block in the middle of the screen.
0020                        *  The user can control it using the Right Joystick.  The four
0021                        *  cardinal directions are supported-- diagnals are NOT supported.
0022                        *
0023                        *  Version
0024                        *  -------
0025                        *  .02 - Added Light-Blue "Walls" to all four sides of the screen
0026                        *      - Added Boundary Checking (White Block can't move past "walls")
0027                        *
0028                        *  .01 - First Release
0029
0030                        *  Equates (Static)
0031
0032 02ef                  MIDSCR    EQU  $02EF        ; Location of Middle of Screen
0033 0200                  TOPLEFT   EQU  $0200        ; Top-Left Corner of Screen
0034 021f                  TOPRIGHT  EQU  $021F        ; Top-Right Corner of Screen
0035 03e0                  LOWRLEFT  EQU  $03E0        ; Lower-Left Corner of Screen
0036 004e                  NORTH     EQU  $4E          ; ASCII Capital N
0037 0053                  SOUTH     EQU  $53          ; ASCII Capital S
0038 0045                  EAST      EQU  $45          ; ASCII Capital E
0039 0057                  WEST      EQU  $57          ; ASCII Capital W
0040 0080                  BACKGRND  EQU  $80          ; ASCII of Background Color
0041 00cf                  WHITEBLK  EQU  $CF          ; ASCII of White Block
0042 00df                  WALL      EQU  $DF          ; ASCII of Light-Blue Block ("Wall")
0043
0044                        *  Equates (RAM Locations - Variables)
0045
0046 01f2                  JOYDATA   EQU  $01F2        ; Joystick Data
0047 0180                  NEWLOC    EQU  $0180        ; New Location of White Block (Two Bytes)
0048 0182                  CURLOC    EQU  $0182        ; Cur Location of White Block (Two Bytes)
0049 0184                  OLDLOC    EQU  $0184        ; Old Location of White Block (Two Bytes)
0050 0186                  ENDCOLUM  EQU  $0186        ; End of Column (Two Bytes)
0051
0052                        *  Equates (On-Board Routines)
0053
0054 4296                  FILLSCRN  EQU  $4296        ; Fill Screen Routine
0055 41d9                  READJOY2  EQU  $41D9        ; Read Right Joystick
0056
0057 8000                           ORG   $8000       ; Start of APF Cartridge Area
0058
0059                        *  The BIOS uses the first five bytes of the cartridge
0060
0061 8000 bb                        FCB   $BB         ; Tell BIOS a cart is present
0062 8001 80 d8                     FDB   MENUSTR     ; Points to Menu string on cartridge
0063 8003 31                        FCB   $31         ; LSB 4 bits is # of Menu Choices
0064                        *                           Menu has one choice
0065 8004 00                        FCB   $00         ; Checksum-type byte.  Must be $00 else
0066                        *                           BIOS restarts right after Init
0067
```

```
0068                        *  After a menu choice is made from the APF Menu via the
0069                        *  BIOS, then control passes to the program here ($8005).
0070
0071 8005 bd 42 96                   JSR   FILLSCRN    ; Fill Screen Background with Black
0072
0073                        **********************
0074                        *                    *
0075                        *  Main Program Loop *
0076                        *                    *
0077                        **********************
0078
0079 8008 bd 80 11                   JSR   INITPROG    ; Initialize Program
0080 800b bd 80 45          MOVBLOCK JSR   READSTIK    ; Read Joystick and Move Block
0081 800e 7e 80 0b                   JMP   MOVBLOCK    ; Continue Running Program Forever
0082
0083                        *****************
0084                        *               *
0085                        *  Subroutines  *
0086                        *               *
0087                        *****************
0088
0089                        *  Initialize Program
0090                        *  ------------------
0091                        *  1) Fill top and bottom rows with Light-Blue (for "Wall")
0092                        *  2) Fill Left and Right side with Light-Blue (for "Wall")
0093                        *  3) Move the White Block to the middle of the screen.
0094
0095                        INITPROG
0096                        *  Fill Top Row
0097 8011 86 df                      LDAA  #WALL       ; Set-Up Color for Walls
0098 8013 ce 02 00                   LDX   #TOPLEFT    ; Prepare for Top-Line Fill Row
0099 8016 bd 80 bd                   JSR   FILLROW     ; Fill the Top Row with Light-Blue
0100                        *  Fill Bottom Row
0101 8019 ce 03 e0                   LDX   #LOWRLEFT   ; Prepare for Top-Line Fill Row
0102 801c bd 80 bd                   JSR   FILLROW     ; Fill the Top Row with Light-Blue
0103                        *  Fill Left-Side Column
0104 801f ce 03 e0                   LDX   #$03E0      ; Bottom of Column Address
0105 8022 ff 01 86                   STX   ENDCOLUM    ; Store at End of Column
0106 8025 ce 02 00                   LDX   #TOPLEFT    ; Top of Column Address
0107 8028 bd 80 c6                   JSR   FILLCOLM    ; Fill Vertical Column
0108                        *  Fill Right-Side Column
0109 802b ce 03 ff                   LDX   #$03FF      ; Bottom of Column Address
0110 802e ff 01 86                   STX   ENDCOLUM    ; Store at End of Column
0111 8031 ce 02 1f                   LDX   #TOPRIGHT   ; Top of Column Address
0112 8034 bd 80 c6                   JSR   FILLCOLM    ; Fill Vertical Column
0113                        * Put White Block in Middle of Screen
0114 8037 ce 02 ef                   LDX   #MIDSCR     ; Middle of Screen
0115 803a ff 01 82                   STX   CURLOC      ; Current location of the White Block
0116 803d ff 01 84                   STX   OLDLOC      ; First Time through Old and Current are same
0117 8040 86 cf                      LDAA  #WHITEBLK   ; Place ASCII White Block into Accum A
0118 8042 a7 00                      STAA  $00,X       ; Store White Block on Screen
0119 8044 39                         RTS
0120
0121                        *  Has the Joystick been moved?
0122                        *  ----------------------------
0123
0124 8045 bd 41 d9          READSTIK JSR   READJOY2    ; Right Joystick Subroutine
0125 8048 25 01                      BCS   JOY2USED    ; Branch if Joystick Used
0126 804a 39                         RTS               ; No Movement- Return and keep checking
0127
0128                        *  Check which direction the joystick has been moved
0129                        *  -------------------------------------------------
0130
0131                        * Check North
0132 804b b6 01 f2          JOY2USED LDAA  JOYDATA     ; Put Joystick Direction Data into Accum A
0133
0134 804e 81 4e                      CMPA  #NORTH      ; Did Joystick Move North?
```

```
0135 8050 26 03                    BNE   CHKSOUTH    ; If not, then Check South Movement
0136 8052 7e 80 7f                 JMP   MOVNORTH    ; Direction was North, Branch
0137
0138                       * Check South
0139 8055 81 53            CHKSOUTH  CMPA  #SOUTH      ; Did Joystick Move South?
0140 8057 26 03                    BNE   CHKEAST     ; If not, then Check East Movement
0141 8059 7e 80 8e                 JMP   MOVSOUTH    ; Direction was South, Branch
0142
0143                       * Check East
0144 805c 81 45            CHKEAST   CMPA  #EAST       ; Did Joystick Move East?
0145 805e 26 03                    BNE   CHKWEST     ; If not, then Check West Movement
0146 8060 7e 80 6b                 JMP   MOVEAST     ; Direction was East, Branch
0147
0148                       * Check West
0149 8063 81 57            CHKWEST   CMPA  #WEST       ; Did Joystick Move West?
0150 8065 26 03                    BNE   NODIR       ; If not, then No Direction was Picked
0151 8067 7e 80 75                 JMP   MOVWEST     ; Direction was West, Branch
0152
0153 806a 39               NODIR     RTS             ; No Movement- Return and keep checking
0154
0155                       * Update White Block's Movement on the Screen
0156                       * ----------------------------------------
0157
0158                       * White Block's East Movement
0159 806b fe 01 82         MOVEAST   LDX   CURLOC      ; Load Current Location of White Block
0160 806e ff 01 84                 STX   OLDLOC      ; Location of White Block BEFORE Update
0161 8071 08                       INX               ; Move Block Location East
0162 8072 7e 80 9a                 JMP   DRAWBLOK    ; Jump to Draw White Block Subroutine
0163
0164                       * White Block's West Movement
0165 8075 fe 01 82         MOVWEST   LDX   CURLOC      ; Load Current Location of White Block
0166 8078 ff 01 84                 STX   OLDLOC      ; Location of White Block BEFORE Update
0167 807b 09                       DEX               ; Move Block Location West
0168 807c 7e 80 9a                 JMP   DRAWBLOK    ; Jump to Draw White Block Subroutine
0169
0170                       * White Block's North Movement
0171 807f fe 01 82         MOVNORTH  LDX   CURLOC      ; Load Current Location of White Block
0172 8082 ff 01 84                 STX   OLDLOC      ; Location of White Block BEFORE Update
0173 8085 86 20                    LDAA  #32D        ; Initialize Loop for 32 Times
0174 8087 09               BLOCKUP   DEX               ; Countdown Loop for "Up" Movement
0175 8088 4a                       DECA              ; Decrement "Up" Movement Loop
0176 8089 26 fc                    BNE   BLOCKUP     ; Has screen RAM moved "Up"
0177 808b 7e 80 9a                 JMP   DRAWBLOK    ; Jump to Draw White Block Subroutine
0178
0179                       * White Block's South Movement
0180 808e fe 01 82         MOVSOUTH  LDX   CURLOC      ; Load Current Location of White Block
0181 8091 ff 01 84                 STX   OLDLOC      ; Location of White Block BEFORE Update
0182 8094 86 20                    LDAA  #32D        ; Initialize Loop for 32 Times
0183 8096 08               BLOKDOWN  INX               ; Countdown Loop for "Down" Movement
0184 8097 4a                       DECA              ; Decrement "Down" Movement Loop
0185 8098 26 fc                    BNE   BLOKDOWN    ; Has screen RAM moved "Down"
0186
0187                       *  Draw White Block - Draw Block at New Locaction.
0188                       *  ----------------
0189                       *  1) Check boundaries.  Has a wall been hit?
0190                       *  2) Draw White Block if no wall was hit
0191
0192                       DRAWBLOK
0193                       *  Boundary Check
0194 809a ff 01 80                 STX   NEWLOC      ; Update New Location of White Block
0195 809d a6 00                    LDAA  $00,X       ; Load What is in Updated Location
0196 809f 81 df                    CMPA  #WALL       ; Is the Updated Location a Wall?
0197 80a1 27 0f                    BEQ   DELYLOOP    ; If Yes, then Don't Draw Block
0198                       *  Draw Block if not a wall
0199 80a3 fe 01 84                 LDX   OLDLOC      ; Not a Wall, so Write Background
0200 80a6 a7 00                    STAA  $00,X       ; Store Background Color onto screen
0201 80a8 fe 01 80                 LDX   NEWLOC      ; Get Current Location of White Block
```

```
0202 80ab 86 cf                        LDAA  #WHITEBLK   ; Place ASCII White Block into Accum A
0203 80ad a7 00                        STAA  $00,X       ; Draw White Block to Screen
0204 80af ff 01 82                     STX   CURLOC      ; New Location is now Current Location
0205
0206 80b2 c6 32              DELYLOOP  LDAB  #50D         ; Number of Outer Loops
0207 80b4 86 ff              OUTLOOP   LDAA  #255D        ; Number of Inner Loops
0208 80b6 4a                 INLOOP    DECA               ; Decrement Inner Loop
0209 80b7 26 fd                        BNE   INLOOP       ; All Inner Loops Complete?
0210 80b9 5a                           DECB               ; Decrement Outer Loop
0211 80ba 26 f8                        BNE   OUTLOOP      ; All Outer Loops Complete?
0212
0213 80bc 39                           RTS                ; Check for More Joystick Movement
0214
0215                         *  Fill a complete Horizontal Line with a character
0216                         *  Accum A = Character to fill row with
0217                         *        X = Where to Start Placement of character
0218
0219 80bd c6 20              FILLROW   LDAB  #32D         ; Set-up Loop for one full line
0220 80bf a7 00              FILLR     STAA  $00,X        ; Place a character on screen
0221 80c1 08                           INX                ; Move Screen Location to Right
0222 80c2 5a                           DECB               ; Has the loops run 32 times?
0223 80c3 26 fa                        BNE   FILLR        ; Continue Filling row if not Zero
0224 80c5 39                           RTS
0225
0226                         *  Fill Column
0227                         *  -----------
0228                         *
0229                         *  Fill a complete Vertical Column with a character
0230                         *  Accum A = Character to fill row with
0231                         *        X = Where to Start Placement of character
0232
0233                         FILLCOLM
0234 80c6 a7 00                        STAA  $00,X       ; Store Character onto screen
0235 80c8 c6 20              COLUMND1  LDAB  #32D         ; Initialize Loop for 32 Times
0236 80ca 08                 COLUMND2  INX                ; Countdown Loop for "Down" Movement
0237 80cb 5a                           DECB               ; Decrement "Down" Movement Loop
0238 80cc 26 fc                        BNE   COLUMND2     ; Has Column Moved Down?
0239 80ce a7 00                        STAA  $00,X        ; Store Character onto screen
0240 80d0 bc 01 86                     CPX   ENDCOLUM
0241 80d3 26 f3                        BNE   COLUMND1
0242 80d5 a7 00                        STAA  $00,X        ; Store Character onto screen
0243 80d7 39                           RTS
0244
0245
0246                         *  Cartridge Menu String
0247                         *  ---------------------
0248                         *
0249                         *  The data required by the BIOS for the main menu is here.
0250
0251 80d8 e9                 MENUSTR   FCB   $E9          ; 9 spaces before Program Name string
0252 80d9 4d 4f 56 45 20 42            FCC   "MOVE BLOCK  .02"  ; Program Name String
     4c 4f 43 4b 20 20
     2e 30 32
0253 80e8 e8                           FCB   $E8          ; 8 spaces after text (EOL)
0254 80e9 e1                           FCB   $E1          ; 1 spaces (Before Author)
0255 80ea 42 59 20 41 44 41            FCC   "BY ADAM TRIONFO, JULY 20, 2010"  ; Author
     4d 20 54 52 49 4f
     4e 46 4f 2c 20 4a
     55 4c 59 20 32 30
     2c 20 32 30 31 30
0256 8108 e3                           FCB   $E3          ; 3 spaces (After Author)
0257 8109 e3                           FCB   $E3          ; 3 spaces (Before Author)
0258 810a 42 41 53 45 44 20            FCC    "BASED ON ERIC BECKETT'S"
     4f 4e 20 45 52 49
     43 20 42 45 43 4b
     45 54 54 27 53
0259 8121 e7                           FCB   $E7          ; 7 spaces (After Author)
```

```
0260 8122 42 41 53 49 43 2f           FCC    "BASIC/ML EXAMPLE FROM 1984"
     4d 4c 20 45 58 41
     4d 50 4c 45 20 46
     52 4f 4d 20 31 39
     38 34
0261 813c fe                          FCB   $FE         ; Another 30 spaces
0262 813d e5                          FCB   $E5         ; Another 5 spaces
0263 813e 31 2e 20 52 55 4e           FCC   "1. RUN PROGRAM                  "
     20 50 52 4f 47 52
     41 4d 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20
0264 815e ff                          FCB   $FF         ; Control Byte - End of String
0265
0266 815f ff ff ff ff ff ff  CARTEND  FILL  $FF,$9000-CARTEND ; Pad with $FF for a 4K Cart
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff ff ff
     ff ff ff ff
0267
0268                        * End of Program
```

```
BACKGRND   0080 *0040
BLOCKUP    8087 *0174 0176
BLOKDOWN   8096 *0183 0185
CARTEND    815f *0266 0266
CHKEAST    805c *0144 0140
CHKSOUTH   8055 *0139 0135
CHKWEST    8063 *0149 0145
COLUMND1   80c8 *0235 0241
COLUMND2   80ca *0236 0238
CURLOC     0182 *0048 0115 0159 0165 0171 0180 0204
DELYLOOP   80b2 *0206 0197
DRAWBLOK   809a *0192 0162 0168 0177
EAST       0045 *0038 0144
ENDCOLUM   0186 *0050 0105 0110 0240
FILLCOLM   80c6 *0233 0107 0112
FILLR      80bf *0220 0223
FILLROW    80bd *0219 0099 0102
FILLSCRN   4296 *0054 0071
INITPROG   8011 *0095 0079
INLOOP     80b6 *0208 0209
JOY2USED   804b *0132 0125
JOYDATA    01f2 *0046 0132
LOWRLEFT   03e0 *0035 0101
MENUSTR    80d8 *0251 0062
MIDSCR     02ef *0032 0114
MOVBLOCK   800b *0080 0081
MOVEAST    806b *0159 0146
MOVNORTH   807f *0171 0136
MOVSOUTH   808e *0180 0141
MOVWEST    8075 *0165 0151
NEWLOC     0180 *0047 0194 0201
NODIR      806a *0153 0150
NORTH      004e *0036 0134
OLDLOC     0184 *0049 0116 0160 0166 0172 0181 0199
OUTLOOP    80b4 *0207 0211
READJOY2   41d9 *0055 0124
READSTIK   8045 *0124 0080
SOUTH      0053 *0037 0139
TOPLEFT    0200 *0033 0098 0106
TOPRIGHT   021f *0034 0111
WALL       00df *0042 0097 0196
WEST       0057 *0039 0149
WHITEBLK   00cf *0041 0117 0202
```