Michael Matte's

Bally Arcade/Astrocade *Gunfight* Assembly Language Breakdown

Overview and Q&A
By Adam Trionfo
May 18, 2017

This document includes:

1) **Overview and Q&A** - An introduction and overview of the Z80 machine language breakdown of *Gunfight*. Also included is some background material asked of Michael Matte in a Q&A format.

2) **Gunfight Breakdown (Typed)** - The first two pages of the breakdown have been typed and are included to make the pdf document friendlier to Internet search engines.

3) **Handwritten *Gunfight* Breakdown** - Michael Matte's complete, 42-page breakdown of the Astrocade game *Gunfight*. This breakdown will be most useful if used with the source code for the Bally's 8K system ROM, which is available in the "Nutting Manual."

4) **Errata Sheet** - Two pages of corrections for errors and/or omissions made in the handwritten document.

Overview

The Bally Arcade/Astrocade game system was released in January 1978 by Bally. It was re-released in 1981 by Astrovision, Inc. The system has an 8K ROM with four built-in programs: *Gunfight*, *Checkmate*, *Calculator* and *Scribbling*. *Gunfight* is a home port of the B&W arcade game *Gunfight*, released by Midway in November 1975. The original release of the game was called *Western Gun: i*t was released by Taito in Japan and used, as was the convention at the time, discrete logic (i.e. the system didn't use a CPU). The North American arcade version of *Gunfight* moved away from the original design. It is usually credited as the first arcade game to use a microprocessor (the Intel 8080 CPU). The Midway arcade version of *Gunfight* was programmed by Tom McHugh.

The Astrocade version of *Gunfight* is probably the most sophisticated of the four programs built into the Astrocade. The *Software and Hardware for the Bally Arcade - A Technical Description* (aka "Nutting Manual" and/or "The Handbook of Hardware & Software") has the complete Z80 assembling language source listing for the game. Many people used this manual, which could be purchased through the *Arcadian* newsletter, to learn to program the Bally Arcade/Astrocade.

In the 1980s, Michael Matte, a passionate Astrocade user, used the source listing for the 8K ROM as a basis for his detailed breakdown of *Gunfight*. Michael created the breakdown "to provide beginner assembly or machine language programmers an inside look at the game *Gunfight*. The documentation will reveal how on-board subroutines in the System ROM can be used to execute particular tasks. The 'special routines' listing can be used as a reference source for programming demos or games."

Michael Matte: Questions & Answers

In May of 2017, Michael Matte sent me his 42-page handwritten *Gunfight* breakdown so that it could be archived on BallyAlley.com. The following are a few questions, asked via email, about the creation of the intriguing document.

Adam: When did you write the breakdown of *Gunfight*? How long did it take you?

Michael: "Wrote this breakdown back in the 80's. Don't recall what year I wrote it or how long it took. Must have rewritten it because the breakdown is well organized. Had no word processor or printer at that time."

Adam: Did you ever share it with anyone before now?

Michael: "I wrote it for myself for future reference. I did not share it with anyone. It does provide insight as to how one can use the on-board subroutines to create a demo or game. I feel it does a better job with comments then what I have seen so far in *Gunfight* documentation. I also plan this style of documentation when I breakdown *The Incredible Wizard* with a greater emphasis on comments and details on the many routines utilized to create all that wonderful graphics and animation. The intent of my *IW* breakdown will be to open the door for those interested in developing skills in programming graphics in assembly or machine language and for future reference."

Adam: What was your experience with Z80 machine language when you began the project?

Michael: "I developed my skills prior to writing my *Gunfight* breakdown by attacking the on-board subroutines. I first broke down the UPI [User Program Interface] instruction by instruction using the Nutting Manual's ROM documentation, then broke down a bunch of the on-board sub's with an emphasis on the graphics routines. I documented that effort for future reference and learned a lot. I also broke down some of Andy Guevara's ML work and learned a great deal. Having Bit Fiddler's *MLM* [*Machine Language Manager* cartridge] was great because it allowed me to experiment and write ML routines and then eventually come up with high-res *MLM*."

Adam: What tools did you use to examine the game's code (i.e. the *MLM*)?

Michael: "I have 2 books covering Z80 machine and assembly language which provide detailed info on using the Z80 instructions and other goodies. Having the *MLM* breakdown as part of the *MLM* User Manual plus my previous ML experience allowed me to figure out how *MLM* worked."

Adam: I presume that you based your breakdown on the ROM's Z80 source code that was included with the "Nutting Manual." Is that correct?

Michael: "I did use the Nutting Manual's breakdown on *Gunfight* as a guide and then expanded that info documenting it with a non-traditional style for future reference in creating or breaking down an Astrocade game. After looking at my *Gunfight* breakdown recently, I could have spent more time on it with regard to comments because there are some parts of the breakdown that are vague. However, when I get to the advanced programming examples in my 'In-Depth Look At..' series, I will be presenting a bunch of ML examples using my *Gunfight* breakdown as a guide and could add some more comments (for my use) to it then."

This page intentionally left blank

GUNFIGHT
An Instruction by Instruction Breakdown

An MCM Design Project
Documentation written by Michael Matte

Intent of Documentation

To provide beginner assembly or machine language programmers an inside look at the game GUNFIGHT. The documentation will reveal how on-board subroutines in the System ROM can be used to execute particular tasks. The "special routines" listing can be used as a reference source for programming demos or games.

Page Layout

The typical page layout lists in the center of the page, the Z80 mnemonics, an abbreviated form, to identify what specific Z80 instructions are executed.

To the left of the mnemonics is the actual Z80 operational coding (Op Codes) that reside in ROM.

To the left of the Op Codes are occasional ROM addresses. The ROM addresses are indicated only occasionally to reduce clutter.

To the right of the Z80 mnemonics are multiple comments. The comments are usually to the right of the brackets. The brackets point and group all the necessary Z80 instructions or data necessary to execute the particular task that is commented.

Special Routines

1)	Is it time to show "GAME OVER"?	@ 1A3AH	
2)	What to do when a trigger is pulled	@ 180AH	*
	Blank and/or write a bullet	@ 1BC7H	
	Update bullet vectors	@ 1C4EH	
	What to do when a bullet hits something	@ 1B0CH	
	A cowboy was shot *		
3)	Blank cowboy, display legs, arm, cowboy	@ 1B78H	
	Set up for next legs	@ 1CC0H	
	Knob changed	@ 18B1H	*
4)	Joystick changed	@ 1895H	*
5)	Shut down for a while	@ 188DH	*
6)	Display a timer	@ 17E1H	*
	What to do when timer reaches zero *		

* Used in SENTRY DO IT routine at 1B03H

GUNFIGHT

AN INSTRUCTION BY INSTRUCTION BREAKDOWN

A MCM DESIGN PROJECT
DOCUMENTATION WRITTEN BY MICHAEL MATTE

INTENT OF DOCUMENTATION

TO PROVIDE BEGINNER ASSEMBLY OR MACHINE LANQUAGE PROGRAMMERS AN INSIDE LOOK AT THE GAME "GUNFIGHT". THE DOCUMENTATION WILL REVEAL HOW ON-BOARD SUBROUTINES IN THE SYSTEM ROM CAN BE USED TO EXECUTE PARTICULAR TASKS. THE "SPECIAL ROUTINES" LISTING CAN BE USED AS A REFERENCE SOURCE FOR PROGRAMMING DEMOS OR GAMES.

PAGE LAYOUT

THE TYPICAL PAGE LAYOUT LISTS IN THE CENTER OF THE PAGE, THE Z80 MNEUMONICS, AN ABBREVIATED FORM, TO IDENTIFY WHAT SPECIFIC Z80 INSTRUCTIONS ARE EXECUTED.

TO THE LEFT OF THE MNEMONICS IS THE ACTUAL Z80 OPERATIONAL CODING (OP CODES) THAT RESIDE IN ROM.

TO THE LEFT OF THE OP CODES ARE OCCASIONAL ROM ADDRESSES THE ROM ADDRESSES ARE INDICATED ONLY OCCASIONALLY TO REDUCE CLUTTER.

TO THE RIGHT OF THE Z80 MNEUNOMICS ARE MULTIPLE COMMENTS THE COMMENTS ARE USUALLY TO THE RIGHT OF BRACKETS. THE BRACKETS POINT AND GROUP ALL THE NECESSARY Z80 INSTRUCTIONS OR DATA NECESSARY TO EXECUTE THE PARTICULAR TASK THAT IS COMMENTED.

SPECIAL ROUTINES

① IS IT TIME TO SHOW "GAME OVER"? @ $1A3A_H$

② WHAT TO DO WHEN A TRIGGER IS PULLED @ $180A_H$ *
BLANK AND/OR WRITE A BULLET @ $1BC7_H$
UPDATE BULLET VECTORS @ $1C4E_H$
WHAT TO DO WHEN A BULLET HITS SOMETHING @ $1B0C_H$
A COWBOY WAS SHOT *

③ BLANK COWBOY, DISPLAY LEGS, ARM, COWBOY @ $1B78_H$
SET UP FOR NEXT LEGS @ $1CC0_H$
KNOB CHANGED @ $18B1_H$ *

④ JOYSTICK CHANGED @ 1895_H *

⑤ SHUT DOWN FOR AWHILE @ 188D *

⑥ DISPLAY A TIMER @ $17E1_H$ *
WHAT TO DO WHEN TIMER REACHES ZERO *

MAIN PROGRAM BEGINS @ $17DE_H$ SEE PAGE 18

* USED IN SENTRY. DO IT ROUTINE @ $1B03_H$

DATA BASE LOCATIONS

$4F12_H$] USED TO DETERIME WHEN INTERRUPT #2 IS TO BE USED (@$1C34_H$).

4F14 4F]
4F15 61] NEXT COWBOY/WAGON $\vec{VP}$ UPDATED

4F16 78]
4F17 4F] LAST COWBOY/WAGON $\vec{VP}$ TO UPDATE

$4F18_H$ 20 MR (XOR IT)
VECTOR STATUS
TIME BASE
ΔX_L] ΔX
ΔX_H]
X_L] X
X_H]
01 X CHECKS MASK
ΔY_L] ΔY
ΔY_H]
Y_L] Y
Y_H]
03 Y CHECKS MASK
] MAGIC
] BLANK ADDRESS
] LAST MR VALUE
— 1ST BULLET DATA FOR COWBOY #1

$4F2A_H$ 20 MR (XOR IT)
VECTOR STATUS
TIME BASE
ΔX_L] ΔX
ΔX_H]
X_L] X
X_H]
01 X CHECKS MASK
— 2ND BULLET DATA FOR COWBOY #1

$4F32_H$
ΔY_L] ΔY
ΔY_H]
Y_L] Y
Y_H]
03 Y CHECKS MASK
] MAGIC
] BLANK ADDRESS
] LAST MR VALUE

$4F3C_H$ 60 MR (FLOP, XOR IT)
VECTOR STATUS
TIME BASE
ΔX_L] ΔX
ΔX_H]
X_L] X
X_H]
01 X CHECKS MASK
ΔY_L] ΔY
ΔY_H]
Y_L] Y
Y_H]
03 Y CHECKS MASK
] MAGIC
] BLANK ADDRESS
] LAST MR VALUE
— 1ST BULLET DATA FOR COWBOY #2

$4F4E_H$ MR (XOR IT)
VECTOR STATUS
TIME BASE
ΔX_L
— 2ND BULLET DATA FOR COWBOY #2

4F52$_H$ ΔX_H

X_L] X

X_H]

01 X CHECKS MASK

ΔY_L] ΔY

ΔY_H]

Y_L] Y

Y_H]

03 Y CHECKS MASK

] MAGIC

] BLANK ADDRESS

] LAST MR VALUE

4F60] USED IN ROUTINE @ 1D54$_H$, 1D64$_H$

4F61$_H$ 10 MR (OR IT)

VECTOR STATUS

TIME BASE

ΔX_L] ΔX

ΔX_H]

X_L] X

X_H]

01 X CHECKS MASK

ΔY_L] ΔY

ΔY_H]

Y_L] Y

Y_H]

01 Y CHECKS MASK

] MAGIC

] BLANK ADDRESS

] SHOOTING ARM PATTERN ADDRESS INDEX

] "LOW ORDER ADDRESS-2" LEGS/FALL PATTERN

COWBOY #1 DATA

78] USED IN ROUTINE @ 1D54$_H$, 1D64$_H$

4F78$_H$ 50 MR (FLOP, OR IT)

VECTOR STATUS

TIME BASE

ΔX_L] ΔX

ΔX_H]

X_L] X

X_H]

01 X CHECKS MASK

ΔY_L] ΔY

ΔY_H]

Y_L] Y

Y_H]

01 Y CHECKS MASK

] MAGIC

] BLANK ADDRESS

] SHOOTING ARM PATTERN ADDRESS INDEX

] "LOW ORDER ADDRESS-2" LEGS/FALL PATTERN

COWBOY #2 DATA

DATA BASE LOCATIONS CONT'D

4F8E	USED IN ROUTINE @ $1D54_H$, $1D64_H$	
$4F8F_H$	MR	WAGON DATA
$4F90_H$	VECTOR STATUS	
	TIME BASE	
	ΔX_L	
	ΔX_H	
	X_L	
	X_H	
	X CHECKS MASK	
	ΔY_L	
	ΔY_H	
	Y_L	
	Y_H	
	Y CHECKS MASK	
	MAGIC BLANK ADDRESS	

4FA1	PLYR1 "CACTUS/TREE DISPLAY" COUNTER
4FA2	PLAYER 1 SCORE
4FA3	
4FA5	PLYR2 "CACTUS/TREE DISPLAY" COUNTER
4FA6	PLAYER 2 SCORE
4FA7	

4FDA	# OF BULLETS LEFT - COWBOY #1	
4FDB	↓ - COWBOY #2	
4FDC	TIME LEFT	
$4FDE_H$	BIT 0 SET - A COWBOY WAS SHOT	
4FF4	MAX SCORE TO PLAY TO (INPUTED @ BEGINNING OF GAME)	
4FF5		
$4FF8_H$	GAME STATUS BYTE	BIT 1 SET @ BEGINNI BIT 7 SET-GAME OV

DISPLAY THE TIMER ROUTINE*

```
17E1H  F3           DI                         ]  DECREMENT "AS A BCD" (4FDCH),
       FF           RST 38H                    ]  IF IT IS NONZERO.
       11           SUB#16 - SUCK REQ'D        ]- REPORT THAT LOCATION AS REACHING
       80           REG. C ]- COUNTER/TIMER    ]  ZERO BY SETTING BIT 7 OF 4FFD
                           ]  CHECK MASK
       DD 21 0D 02  LD IX, 020DH                  IX= ADDRESS OF NON-STD
                                                      FONT DESCRIPTOR TABLE
       3A DC 4F     LD A, (4FDCH)              ]   IS (4FDCH)=0?
       B7           OR A                       ]- IF YES, DON'T BOTHER TO DISPLAY TIM
       28 08        JR Z,     IF Z=1, JUMP     ]  IF NOT, DISPLAY THE TIMER
       FF           RST 38H                                          ]
       37           SUB#54 - SUCK REQ'D                              ]     DISPLAY
       4C 02        E, D REG. ]- X,Y COORDINATES                     ]-  THE 2 DIGIT
       0B             C    |  ]- CHAR DISPLAY PARAMETERS             ]     TIMER
       42             B    |  ]- DISPLAY 2 NON-STD CHAR'S            ]
       DC 4F          L, H v  ]- DISPLAY BCD'S @ 4FDCH               ]
       AF           XOR                        ]
       D3 0C        OUT (0CH), A               ]-  ZERO 4FFFH ?
       32 FF 0F     LD (0FFFH), A              ]
       FB           EI
17FE   C9           RET
```

* USED BY DO IT ROUTINE @ $1B08_H$

TRIGGER CHANGED ROUTINE *

```
17FFH  FF        RST 38H                                   ⎤ SACRED REGISTER SUCK
       0D        SUB#12 - REG LOAD REQ'D PER NEXT BYTE     ⎥ IX= COWBOY #1 V⃗P ADDRESS
       DC        SUCK H,L,IX,B,C   1101 1100               ⎥ BC→POINTS TO # OF COWBOY #1
       61 4F       IXL, IXH REG.   | IX=4F61               ⎥       BULLETS LEFT
       DA 4F          C,B          | BC=4FDA               ⎥ HL→ POINTS TO 1ST BULLET VECTOR ST
       19 4F          L,H    ↓     | HL = 4F19             ⎦   FOR COWBOY #1.
       1809         JR                                     JUMP TO 1813H
180AH  FF        RST 38H                                   ⎤ SACRED REGISTER SUCK
       0D        SUB#12 - REG LOAD REQ'D PER NEXT BYTE     ⎥ IX= COWBOY #2 V⃗P ADDRESS
       DC        SUCK H,L,IX,B,C   1101 1100               ⎥ BC→ POINTS TO # OF COWBOY #2
       78 4F       IXL, IXH REG.   | IX= 4F78              ⎥       BULLETS LEFT
       DB 4F          C,B          | BC = 4FDB             ⎥ HL → POINTS TO 1ST BULLET VECTOR ST
       3D 4F          L,H    ↓     | HL= 4F3DH             ⎦   FOR COWBOY#2.
1813H  FD 7E 07  LD A,(IY+07)                              PUT TRIGGER/JOYSTICK STATUS IN A
       B7         OR A                                     ⎤ WAS TRIGGER PULLED?
       C8         RET Z    IF Z=1, RETURN                  ⎥ IF YES, CONTINUE.
                                                           ⎦ IF NOT, RETURN
       0A         LD A,(BC)                                ⎤ ARE WE OUT OF BULLETS?
       B7         OR A                                     ⎥ IF YES, RETURN
       C8         RET Z                                    ⎦ IF NOT, CONTINUE
       7E         LD A,(HL)                                ⎤ IS INITIALIZATION OF 1ST BULLET V⃗P
       B7         OR A                                     ⎥ REQ'D, IE, VECTOR STATUS =0?
                                                           ⎥ IF NO, CONTINUE (CHECK 2ND V⃗P)
       2809       JR Z,    IF Z=1, JUMP                    ⎦ IF YES, JUMP (NO NEED TO CHECK 2ND V⃗
       111200     LD DE,0012                               ⎤ POINT HL AT 2ND BULLET V⃗P
       19         ADD HL,DE                                ⎦ VECTOR STATUS.
       7E         LD A, (HL)                               ⎤ DOES THIS V⃗P REQUIRE INITIALIZATIO
       B7         OR A                                     ⎥ IF YES, SKIP NEXT INSTRUCTION
       2801       JR Z,                                    ⎦ IF NO, RETURN (NO INITIALIZING REQ'D.
       C9         RET
       0A         LD A,(BC)                                ⎤ GET THE # OF BULLETS
       3D         DEC A        Z=1 ON ZERO RESULT          ⎥ DECREMENT THIS #
       02         LD (BC),A                                ⎦ PUT THIS NEW NUMBER BACK
```

* USED BY DO IT ROUTINE @1B08H

```
182BH  20 0D        JR NZ,          IF Z=0,     ] ARE THERE STILL BULLETS LEFT?
                                    JUMP        ] IF YES, NO NEED TO SET TIMER
       3A DC 4F     LD A,(4FDCH)                ] IS THE TIME LEFT =0?
       B7           OR A            Z=1 ON      ] IF YES, START THE COUNTDOWN, ie,
                                    ZERO RESULT ] SET THE TIMER TO 10H.
       3E 10        LD A,10H                    ]
       28 02        JR Z,           IF Z=1, JUMP] IF NO, WE'RE ALREADY COUNTING
       3E 02        LD A,02H                    ] DOWN. SET THE TIMER TO 02H.
       32 DC 4F     LD (4FDCH),A                ]
       E5           PUSH HL                     ]         SAVE
                                                ] BULLET VECTOR STATUS POINTER
       DD E5        PUSH IX                     ] COWBOY VP POINTER
       0A           LD A,(BC)                   ] PUT THE # OF                 ]
       6F           LD L,A                      ] BULLETS LEFT                 ]
       26 00        LD H,00H                    ] IN HL                        ]
       29           ADD HL,HL                   ] CALCULATE                    ]
       29           ADD HL,HL                   ] THE ΔX                       ]
                                                ] INCREMENT                    ]
       11 68 02     LD DE,0268H        DE = Y,X COORDINATES OF                 ]
                                            1ST BULLET (COWBOY #2)             ]
       DD CB 00 76  BIT 6,(IX)                  ] SET THE FLOP BIT             ]
       3E 40        LD A,40H                    ] FOR THE MAGIC REG            ]
       28 01        JR Z,   IF Z=1,             ] (IF REQ'D)                   ] BLANK THE
                            JUMP                ]                              ] BULLET (AT
       AF           XOR A                       ]                              ] TOP OF SCREE
       19           ADD HL,DE                   ] DE=Y,X COORDINATES           ] USED.
       EB           EX DE,HL                    ]    OF BULLET TO BE           ]
                                                ]    BLANKED                   ]
       FF           RST 38H                     ] CALCULATE THIS               ]
       3A           SUB# 58 -  NO SUCK          ] BULLETS                      ]
                               REQ'D            ] NONMAGIC ADDRESS             ]
       EB           EX DE,HL           PUT THIS NONMAGIC ADDRESS IN HL         ]
       06 05        LD B,05H                    ]                              ]
       11 28 00     LD DE,0028H                 ] BLANK THIS                   ]
       36 FF        LD (HL),FF                  ] BULLET                       ]
       19           ADD HL,DE                   ] (5 LINES HIGH)               ]
       10 FB        DJNZ                        ]                              ]
       16 00        LD D,00H                                 ]
       DD 5E 0F     LD E,(IX+0FH)  E = SHOOTING ARM          ] DE= ADDRESS OF
                                       INDEX                 ]     BULLET VP
       62           LD H,D                                   ]     INITIAL VALUES
       6B           LD L,E                                   ]     (TABLE BEGINS
1865H  29           ADD HL,HL                                ]     @1D8FH)
```

TRIGGER CHANGED ROUTINE CONT'D

```
1866H  19        ADD HL,DE
       11 8F1D   LD DE,1D8FH
       19        ADD HL,DE
       EB        EX DE,HL
       C1        POP BC                BC = COWBOY VP ADDRESS
       E1        POP HL
       E5        PUSH HL               HL POINTS AT BULLET VECTOR STATU:
       23        INC HL
       36 01     LD (HL),01            SET TIME BASE TO 01H
       23        INC HL                HL POINTS AT BULLET ΔXL
       03        INC BC
       03        INC BC                BC POINTS AT COWBOY ΔXL
       03        INC BC
       CD D3 19  CALL ROUTINE @19D3H   INITIALIZE BULLET ΔXL, ΔXH, XL, XH
                                       EXIT WITH: HL POINTING AT BULLET XH
       03        INC BC                           BC POINTING AT COWBOY XH
       03        INC BC                BC POINTS AT COWBOY ΔYL
       23        INC HL                HL POINTS AT BULLET X CHECKS MASK
       36 01     LD (HL),01            SET BULLET X CHECKS MASK TO 01H.
       23        INC HL                HL POINTS AT BULLET ΔYL
       CD D3 19  CALL ROUTINE @19D3H   INITIALIZE BULLET ΔYL, ΔYH, YL, YH
       E1        POP HL                SET BULLET VECTOR STATUS TO 80H
       36 80     LD (HL),80H
       FF        RST 38H
       13        SUB#18 - SUCK REQ'D   TIME FOR THE
       12 4F     IXL, IXH REG.         GUNSHOT SOUND
       01          A
       D7 1F       L,H
188CH  C9        RET
```

TIME TO SHUT DOWN *

```
188D_H  48           SUB#72 - NO SUCK REQ'D                SHUT THE SCREEN DOWN
        09           SUB# 8 - NO ARGUMENTS WILL BE LOADED  ABORT THE NEST
```

JOYSTICK CHANGED ROUTINE *

```
188F_H  DD 21 61 4F   LD IX, 4F61_H          IX= COWBOY #1 V⃗P ADDRESS
        18 04         JR                     SKIP NEXT INSTRUCTION
1895_H  DD 21 78 4F   LD IX, 4F78_H          IX= COWBOY #2 V⃗P ADDRESS
        DD 4E 00      LD C, (IX)
        11 80 00      LD DE, 0080_H          DE= CRT RIGHT MOVEMENT INCREME
        21 80 00      LD HL, 0080_H          HL= CRT DOWN MOVEMENT INCREMEN
        FF            RST 38_H              ┐
        7E            SUB#126- NO SUCK      ├ MASK THE DELTA'S
                               REQ'D        ┘
        DD 74 09      LD (IX+09), H         ┐ PUT MASKED ΔY IN
        DD 75 08      LD (IX+08), L         ┘  COWBOY V⃗P
        DD 72 04      LD (IX+04), D         ┐ PUT MASKED ΔX IN
        DD 73 03      LD (IX+03), E         ┘ COWBOY V⃗P
18B0_H  C9            RET
```

KNOB CHANGED ROUTINE *

```
18B1_H  DD 21 78 4F   LD IX, 4F78_H    IX= COWBOY #2 V⃗P ADDRESS
        78            LD A, B          PUT NEW KNOB VALUE IN A
        2F            CPL              INVERT THE VALUE. COWBOY #2 ARM
                                       MOTION WITH RESPECT TO KNOB MOTION IS
                                       OPPOSITE THAT OF COWBOY #1.

        18 05         JR               SKIP NEXT 2 INSTRUCTIONS
18B9_H  DD 21 61 4F   LD IX, 4F61_H    IX= COWBOY #1 V⃗P ADDRESS
        78            LD A, B          PUT NEW KNOB VALUE IN A
        E6 E0         AND E0_H         ISOLATE BITS 5,6&7
```

* USED BY DO IT ROUTINE @1B08$_H$

```
18C0_H  0F         RRCA                          ]  SHIFT ISOLATED BITS
        0F         RRCA                          ]  INTO BITS 1,2&3
        0F         RRCA                          ]  RESPECTIVELY
        0F         RRCA                          ]
                                   Z=0, IF       ]  IS A = 14_D ?
        FE 0E      CP 0E_H         A≠0E_H        ]  IF YES, SET A TO 12_D.
        20 02      JR NZ,     IF Z=0, JUMP       ]  IF NOT, LEAVE A ALONE.
        3E 0C      LD A, 0C_H                    ]
        DD 77 0F   LD (IX+0F_H), A                  SAVE THE SHOOTING ARM INDEX*
18CD_H  C9         RET
```

** POSSIBLE INDEXES = 0,2,4,6,8,10,1

BULLET HIT SOMETHING ROUTINE

ENTER WITH IX = BULLET VECTOR PACKET ADDRESS

```
18CEH  DD 7E 01      LD A,(IX+01)                       ]  PUT THE BULLET VECTOR STATUS IN
       E6 60         AND 60H                            ]  ISOLATE BITS 5&6
       FE 20         CP 20H    Z=1, IF ONLY BIT 5       ]  WAS THE BULLET WRITTEN OVER
                               IS SET                      SOMETHING AND THEN BLANKED?
                                                           IF YES, JUMP.            ↖ BY
       28 0F         JR Z,     IF Z=1, JUMP             ]  IF NOT, CONTINUE.   INTERRUPT ROU
                                                                                      #1
       D0            RET NC    IF C=0, RET              ]  CONTINUE IF BULLET WASN'T WRITTEN O
                                                           SOMETHING AND WAS BLANKED. OTHERWISE, R
       DD CB 07 5E   BIT 3,(IX+07)                      ]  HAS THE BULLET REACHED ITS X LIMI
                                                           IF YES, CONTINUE.
       C8            RET Z     IF Z=1, RET              ]  IF NO, RETURN
       DD 36 01 00   LD (IX+01),00H                        ZERO THE VECTOR STATUS.
       DD 36 07 01   LD (IX+07),01H                        TURN OFF THE X CHECKS MASK
                                                           LIMIT ATTAINED BIT.
       C9            RET
       DD 7E 06      LD A,(IX+06H)                      ]  IS BULLET PASSED 1ST COLUMN
                                                           OF CACTUS/TREE?
       FE 48         CP 48H    C=0, IF A≥48H            ]  IF YES, JUMP
                                                           IF NO, CONTINUE
       30 0E         JR NC,    IF C=0, JUMP             ]
       DD 36 02 02   LD (IX+02),02H                        SET TIME BASE TO 02H
       DD 36 01 80   LD (IX+01),80H                        TURN ON ACTIVE BIT, KILL BIT 5
       21 8B 1D      LD HL,1D8BH                           HL = ADDRESS OF BULLET LIMIT TAB
       FF            RST 38H                            ]      UPDATE
       3E            SUB#62 - NO SUCK                   ]    BULLET VECTOR
                              REG'D
       C9            RETURN
       DD 36 01 00   LD (IX+01),00H                        ZERO THE VECTOR STATUS
                                                                                        CACT
       FE 58         CP 58H    C=0, IF A≥58H            ]  HAS BULLET TOUCHED 2ND COLUMN OF TREE
                                                           IF YES, JUMP.
       30 1D         JR NC,    IF C=0, JUMP             ]  IF NO, CONTINUE.
       3A 90 4F      LD A,(4F90H)                       ]  IS THE WAGON PRESENT?
       B7            OR A      Z=1, IF (4F90H)=0        ]  IF YES, RETURN.
                                                           IF NO, CONTINUE - CENTER CACTUS IS
       C0            RET NZ    IF Z=0, RET              ]  PRESENT.
       1E 4C         LD E,4CH                              E = X COORDINATE (SAME AS CACTUS
190AH  DD 56 0B      LD D,(IX+0BH)                      ]  D = BULLET Y POSITION - 1
       15            DEC D                              ]
       FF            RST 38H                            ]
       3B            SUB#58 - SUCK REG'D                ]  DETERMINE SCREEN ADDRESS
1910H  00            A REG. ] MR                        ]
```

```
              1911H  EB            EX DE,HL                            HL= SCREEN ADDRESS
                     11 D7FF       LD DE, FFD7H                        DE = -29D
                     06 00         LD B,00                             PUT CONTENTS OF THAT SCREEN
BLANK                7E            LD A,(HL)                           ADDRESS IN A REG.
PORTION              70            LD (HL),B                           BLANK THAT SCREEN ADDRESS
ABOVE                23            INC HL                              POINT TO NEXT BYTE ON SCREEN
CACTUS/TREE          B6            OR (HL)                             OR IT WITH PREVIOUS BYTE
WHERE                70            LD (HL),B                           BLANK THAT "NEXT BYTE" ON SCREEN
BULLET               19     -8     ADD HL,DE                           HL POINTS TO NEXT LINE "ABOVE" TO BLAN
TOUCHED.             20 F8         JR NZ,        IF Z=0, JUMP          IS THERE MORE ABOVE TO BLANK?
                                                                       LOOP BACK IF SO. OTHERWISE RETURN.
                     C9            RET
                     FE 60         CP 60H        C=0, IF A≥60H         IS BULLET PASSED 2ND COLUMN OF CACTU/TREE
                     30 0C         JR NC,        IF C=0, JUMP          IF YES, JUMP
                                                                       IF NO, CONTINUE
                     1E 40         LD E,40H
                     DD CB 00 76   BIT 6,(IX)                          SET E (X COORDINATE) TO 40H OR
                     20 DE  -34    JR NZ,        IF Z=0, JUMP          58H DEPEDING ON WHICH COWBOY'S
                     1E 58         LD E,58H                            BULLET IS BEING EXAMINED. THEN
                     18 DA  -38    JR                                  JUMP TO 190AH TO BLANK THAT PORTION
                                                                       OF CACTUS/TREE.
              1930H  DD CB 00 76   BIT 6,(IX)                          IS FLOP BIT OF BULLET VP
                     28 0C         JR Z,         IF Z=1, JUMP          MAGIC REGISTER SET?
                                                                       IF YES, CONTINUE
                                                                       IF NOT, JUMP TO 1942H
                     FF            RST 38H
                     0D            SUB#12- REG LOAD REQ'D              IX= 4F61  (COWBOY #1 VP ADDRESS
                     DD            SUCK IX, E, C, B, L, H              E = 08  ("GOT ME" X COORDINATE
                     61 4F         IXL, IXH REG.                       BC = 1FAD (MUSIC PARA STRING ADDRE
                     08            E                                   HL = 4FA6 (PLAYER 2 SCORE)
                     AD 1F         C,B
                     A6 4F         L,H
                     18 0A         JR                                  JUMP TO 194CH
              1942H  FF            RST 38H
                     0D            SUB#12- REG LOAD REQ'D              IX= 4F78H (COWBOY #2 VP ADDRESS)
                     DD            SUCK IX, E, C, B, L, H              E = 64H ("GOT ME" X COORDINATE)
                     78 4F         IXL, IXH REG.                       BC = 1FBD (MUSIC PARAM STRING ADDRE
                     64            E                                   HL = 4FA2 (PLAYER 1 SCORE)
                     BD 1F         C,B
                     A2 4F         L,H
```

BULLET HIT SOMETHING ROUTINE CONT

```
194CH DD 361106   LD (IX+11H),06H                   "
      DD 361280   LD (IX+12H),80H                   "USE COWBOY FALLING DOWN FRAME (
      DD 360168   LD (IX+01),68H                    SET COWBOY VECTOR STATUS TO 68H
      DD 7E 0B    LD A,(IX+0BH)                     GET COWBOY Y POSITION, PUT IT IN
      D6 08       SUB 08H                      ⎤ DETERMINE Y COORDINATE
      FE 13       CP 13H   C=0, IF A≥13H       ⎥ FOR "GOT ME" AND PUT IT IN D REG
      30 02       JR NC,  IF C=0, JUMP         ⎥- IF COWBOY IS TO HIGH, SET
      C6 20       ADD A,20H                    ⎥  Y COORDINATE BELOW COWBOY.
      57          LD D,A                       ⎦
1964H FF          RST 38H                      ⎤ INCREMENT COWBOY SCORE
      54          SUB#84- No SUCK REQ'D        ⎥- SET BIT 7 OF 4FF8H IF INCREMENTED
                                               ⎦ SCORE ≥ SCORE TO PLAY TO
      2B          DEC HL                  HL POINTS @ "CACTUS/TREE DISPLAY" COUNTE
      7E          LD A,(HL)                    ⎤
      FE 05       CP 05      IF A<5, C=1       ⎥- INCREMENT THIS COUNTER IF
      CE 00       ADC A,00H                    ⎥    IT IS<5.
      77          LD (HL),A                    ⎦
      60          LD H,B                       ⎤- HL = MUSIC PARAMETER     ⎤
      69          LD L,C                       ⎦      STRING ADDRESS       ⎥ PLAY
      DD 21124F   LD IX,4F12          IX = WORKING AREA                    ⎥ "COWBOY
                                           BEGINNING ADDRESS               ⎥- IS SHOT
      3E C0       LD A,C0H                                                 ⎥ MUSIC
      FF          RST 38H                                                  ⎥
      12          SUB#18- No SUCK REQ'D                                    ⎦
      0E 0C       LD C,0CH      ⎤ CHAR DISPLAY PARAMETERS                  ⎤ KILL THE
      21 021F     LD HL,1F02H   ⎦ STRING ADDRESS                           ⎥ INTERRUP
      F3          DI                                                       ⎥- DIPLAY
      FF          RST 38H                                                  ⎥ "GOT ME"
      34          SUB#52 - No SUCK REQ'D                                   ⎦
      FF          RST 38H                                  ⎤ START UP INTERRUPTS
      51          SUB#80 - SUCK REQ'D                      ⎥-      AGAIN
      FA                                                   ⎦ WAIT A BIT
      3E 01       LD A,01H                                 ⎤ SET BIT 0 OF 4FDEH
      32 DE4F     LD (4FDEH),A                             ⎥- "A COWBOY WAS SHOT"
1987H C9          RET                                      ⎦
```

CACTUS/TREE ROUTINE

ENTER WITH: CACTUS/TREE COUNTER IN A
X COORDINATE IN E
Y COORDINATES TABLE (1DBE$_H$ OR 1DB9$_H$) IN BC

```
1988H 21F41E   LD HL,1EF4H                       HL= CACTUS PATTERN ADDRESS
      F5       PUSH AF                           SAVE THE COUNTER
      3E 08    LD A,08H                        ] EXPAND WITH BROWN ON A
      D3 19    OUT (19H),A                     ] YELLOW BACKGROUND
      F1       POP AF                            PUT THAT COUNTER BACK IN A RE
                                               ] SHOULD A CACTUS/TREE BE DISPLAYED?
      FE 01    CP 01         C=1, IF A<1       ] NO, RETURN (COUNTER=0)
      D8       RET C         IF C=1, RET       ] YES, CONTINUE (COUNTER ≥1)
                                               ] IS COUNTER ≥4?
      FE 04    CP 04         C=0, IF A≥4       ] YES, DO NOT WRITE A CACTUS
      30 03    JR NC,        IF C=0, JUMP      ] NO, WRITE A CACTUS
      CD C819  CALL ROUTINE @ 19C8H              PUT A CACTUS ON THE SCREEN
      03       INC BC                            POINT AT NEXT Y COORDINATE
                                               ] IS COUNTER =1?
      FE 02    CP 02         C=1, IF A<2       ] YES, YOUR DONE-RETURN
      D8       RET C         IF C=1, RET       ] NO, THERE'S MORE TO WRITE
                                               ] IS COUNTER ≥5?
      FE 05    CP 05         C=0, IF A≥5       ] YES, DO NOT WRITE A CACTUS.
      30 03    JR NC,        IF C=0, JUMP      ] NO, WRITE A CACTUS
      CD C819  CALL ROUTINE @ 19C8H              PUT A CACTUS ON THE SCREEN
                                               ] IS COUNTER =2?
      FE 03    CP 03         C=1, IF A<3       ] YES, YOUR DONE- RETURN
      D8       RET C         IF C=1, RET       ] NO, THERE'S MORE TO WRITE
      03       INC BC                            POINT AT NEXT Y COORDINATE
      08       EX AF,AF'                         SAVE THE COUNTER
      3E 81    LD A,81H                        ] SET VECTOR STATUS OF WAGON
      32 904F  LD (4F90H),A                    ] VECTOR PACKET TO 81H.
      08       EX AF,AF'                         PUT THAT COUNTER BACK IN A REG.
      CD C819  CALL ROUTINE @ 19C8H            ] PUT THE MIDDLE CACTUS ON THE SCRE
                                               ] IS COUNTER =3?
      FE 04    CP 04         C=1, IF A<4       ] YES, YOUR DONE - RETURN
      D8       RET C         IF C=1, RET       ] NO, THERE'S MORE TO WRITE
      03       INC BC                            POINT AT NEXT Y COORDINATE
      21 E51D  LD HL,1DE5H                       HL = TREE PATTERN ADDRESS
      F5       PUSH AF                           SAVE THE COUNTER
```

```
19BCH  3E 0C     LD A, 0CH            ] EXPAND WITH GREEN ON
       D3 19     OUT (19H), A         ] A YELLOW BACKGROUND
       F1        POP AF               PUT THAT COUNTER IN A REG.
       CDC819    CALL ROUTINE @19C8H  PUT A TREE ON THE SCREEN
       FE 05     CP 05                ] IS COUNTER = 4?
       D8        RET C                ] YES, YOUR DONE - RETURN
                                      ] NO, THE LAST TREE MUST BE WRITTE
19C7H  03        INC BC               POINT AT LAST Y COORDINATE.
```

AND FALL INTO

↓

CACTUS/TREE WRITE ROUTINE

```
19C8H  F5      PUSH AF                      SAVE THE COUNTER
       D5      PUSH DE
       0A      LD A, (BC)                   ] D=Y COORDINATE
       57      LD D, A                      ]
       3E 08   LD A, 08                     MAGIC REGISTER- EXPAND IT
       FF      RST 38H
       22      SUB#34 - NO SUCK REQ'D       WRITE THE CACTUS/TREE
       D1      POP DE
       F1      POP AF                       PUT THE COUNTER BACK IN A REC
19D2H  C9      RET
```

BULLET ΔX, X (ΔY, Y) INITIALIZATION ROUTINE *

SEE NOTE

```
19D3H  1A        LD A,(DE)
       77        LD (HL),A
       13        INC DE
       03        INC BC
       23        INC HL
       1A        LD A,(DE)
       77        LD (HL),A
       23        INC HL
       13        INC DE
       03        INC BC
       3600      LD (HL),00H
       03        INC BC
       23        INC HL
       0A        LD A,(BC)
       EB        EX DE,HL
       86        ADD A,(HL)
       EB        EX DE,HL
       77        LD (HL),A
       13        INC DE
19E7H  C9        RET
```

NOTE: ENTER THIS ROUTINE WITH: DE = ADDRESS OF BULLET $\vec{VP}$ INITIAL VALUES (TABLE BEGINS @1D8F

HL POINTING TO BULLET ΔX_L (ΔY_L

BC POINTING TO COWBOY ΔX_L (ΔY_L

* USED BY TRIGGER CHANGED ROUTINE @17FF$_H$ (180A$_H$)

MAIN PROGRAM

```
17DEH    C3 E8 19     JP 19E8H

19E8H    FF           RST 38H                                            ]  INPUT "MAX SCOR
         4D           SUB#76- SUCK REQ'D                                 ]  TO PLAY TO" FRO
         1E02         REG. C,B ] USER MESSAGE ADDRESS                    ]  KEYBOARD OR HA
                                 ("MAX SCORE")
         84             |   A  ] NUMBER DISPLAY OPTIONS                  ]  CONTROL #1. SAVE
                        |        (BLANK LEADING O'S- 4 DIGIT INPUT)
         F4 4F          ↓  L,H ] DIGITS INPUTED ARE STORED               ]  INPUT AT 4FF4H
                                 BEGINNING AT THIS ADDRESS
         31 06 4F     LD SP, 4F06H
19F2H    FF           RST 38H
         00           BEGIN MULTIPLE CALLS
         1B           SUB#26- SUCK REQ'D                     ]  ZERO LOCATIONS
         06 4F        REG. E,D ] ADDRESS AT WHICH            ]  4F06 - 4FDBH
                                 TO BEGIN FILLING
         D6 00          |  C,B ] # OF BYTES                  ]
                        |        TO FILL
         00             ↓   A  ] FILL WITH 00                ]
         7B           SUB#122 - SUCK REQ'D                   ]  SET
         02           REG. A ] VALUE OF BYTE                 ]  GAME STATUS BYTE
                               TO SET                           TO 02H
         F8 4F        ↓  L,H ] ADDRESS OF BYTE               ]  (USED BY ROUTINE @ 1964H)
                               TO SET
         17           SUB#22- SUCK REQ'D                     ]  SET
         B8           REG. D ] VERTICAL BLANK REGISTER       ]  VERTICAL BLANK REGISTER TO 1
                               (PORT 10D)
11010110  1A00H  D6     |  B ] HORIZONTAL COLOR              ]  HORIZONTAL COLOR BOUNDARY TO D
                               BOUNDARY (PORT 9)
         08             ↓  A ] INTERRUPT MODE                ]  INTERRUPT MODE TO 8.
                               (PORT 0EH)
         19           SUB#24- SUCK REQ'D                     ]  SET
         C3 1D        L,H REG. ] COLOR SPEC                  ]  COLORS
                                 TABLE ADDRESS
         13           SUB#18 - SUCK REQ'D                    ]  BEGIN
         12 4F        IXL, IXH REG. ] WORK AREA              ]  PLAYING THE
                        |             BEGINNING ADDRESS
         C0           A |                                    ]  INTRODUCTORY
         9F 1F        L,H ↓  ] PARAMETER                     ]  MUSIC
                               STRING ADDRESS
         02           END MULTIPLE CALLS
1A0CH    F3           DI                       KEEP THE 1ST NOTE GOING
```

```
1A0DH   FF       RST 38H
        00       BEGIN MULTIPLE CALLS
        5F       SUB # 94 - SUCK REQ'D                                SET COUNTERS AND
1A10H   DA 4F    E,D REG.  ADDRESS TO BEGIN MOVING TO                 LAST STATUS
        0C 00    C,B       # OF BYTES TO MOVE                         (LOCATIONS 4FDA - 4FE5H)
        CB1D     L,H       ADDRESS TO BEGIN MOVING FROM
        1B       SUB # 26 - SUCK REQ'D                                FILL IN TOP 9 LINES
        00 40    E,D REG.  ADDRESS TO BEGIN FILLING                   GREEN
        68 01    C,B       # OF BYTES TO FILL
        FF       A         FILL WITH FFH
        1B       SUB # 26 - SUCK REQ'D                                FILL IN YELLOW
        68 41    E,D REG.  ADDRESS TO BEGIN FILLING                   "GROUND"
        F8 0C    C,B       # OF BYTES TO FILL
1A21H   00       A         FILL WITH 00H
        1B       SUB # 26 - SUCK REQ'D                                ZERO LOCATIONS
        12 4F    E,D REG.  ADDRESS TO BEGIN FILLING                   4F12 - 4FA0H
        8F 00    C,B       # OF BYTES TO FILL
        00       A         FILL WITH 00H
        0D       SUB # 12 - REG LOAD REQ'D
        10       B REG.    SUCK IX                                    IXCB = 020DH
        0D       IXL       IX = 020DH
        02       IXH
        37       SUB # 54 - SUCK REQ'D
        08 02    E,D REG.  X,Y COORDINATES                            DISPLAY
        0B       C         CHAR DISPLAY PARAMETERS                    PLAYER 1 SCORE
1A30H   C4       B         DISPLAY 4 DIGITS BLANK LEADING ZEROES
        A2 4F    L,H       NIBBLE TABLE ADDRESS
        37       SUB # 54 - SUCK REQ'D
        88 02    E,D REG   X,Y COORDINATES                            DISPLAY
        0B       C         CHAR DISPLAY PARAMETERS                    PLAYER 2 SCORE
        C4       B         DISPLAY 4 DIGITS BLANK LEADING ZEROES
        A6 4F    L,H       NIBBLE TABLE ADDRESS
        05       SUB # 4 - SUCK REQ'D                                 JUMP TO 1B2C,
1A3BH   2C 1B    L,H REG.  ADDRESS TO JUMP TO                         TO SEE IF GAME
                                                                      IS OVER.
```

MAIN PROGRAM CONT'D

```
1A3DH  35            SUB#52 - SUCK REQ'D                               WRITE
       2C 01         E,D REG. } X,Y COORDINATES                   "GET READY"
1A40H  0B            C        } CHAR DISPLAY PARAMETERS           AT TOP OF SCREEN
       7A 1D         L,H      } STRING ADDRESS
       02            END MULTIPLE CALLS
       AF            XOR A                                  ZERO VECTOR STATUS
       32 90 4F      LD (4F90H),A                           IN WAGON VECTOR PACKE
       3A A1 4F      LD A,(4FA1H)        GET PLYR1 CACTUS/TREE COUNTER       DISPLAY TH
       1E 58         LD E,58H            E=X COORDINATE (88D)                CACTUS/TRE
       01 BE 1D      LD BC,1DBEH         BC= ADDRESS OF PLYR1                ON THE
                                             Y COORDINATES TABLE             RIGHT SI
1A50H  CD 88 19      CALL ROUTINE @ 1988H                                     OF SCREEN
       3A A5 4F      LD A,(4FA5)         GET PLYR2 CACTUS/TREE COUNTER       DISPLAY T
       1E 40         LD E,40H            E=X COORDINATE (64D)                CACTUS/TREE
       01 B9 1D      LD BC,1DB9H         BC=ADDRESS OF PLYR2                 ON THE
                                             Y COORDINATES TABLE             LEFT SIDE
       CD 88 19      CALL ROUTINE @1988H                                      OF SCREEN
       3E 4F         LD A,4FH
1A60H  32 14 4F      LD (4F14H),A                    (4F14H) ← 4F
       32 17 4F      LD (4F17H),A                    (4F17H) ← 4F
       DD 21 61 4F   LD IX,4F61H               SET MAGIC REGISTER OF
       DD 36 00 10   LD (IX),10H               COWBOY#1 VECTOR PACKET TO 10,
       21 15 4F      LD HL,4F15H             INITIALIZE COWBOY #1 VP
1A71H  CD 30 1D      CALL ROUTINE @ 1D30H
       DD 21 78 4F   LD IX,4F78H               SET MAGIC REGISTER OF
       DD 36 00 50   LD (IX),50H               COWBOY#2 VECTOR PACKET TO 50
       CD 30 1D      CALL ROUTINE @1D30H     INITIALIZE COWBOY#2 VP
       3A 90 4F      LD A,(4F90H)                     IS VECTOR STATUS OF WAGON
1A82H  B7            OR A      Z=1, IF A=0            VECTOR PACKET =0?
       28 1D         JRZ,      IF Z=1, JUMP           IF YES, JUMP TO WRITE A CACT
                                                      IF NO, CONTINUE.
       DD 21 8F 4F   LD IX,4F8F                       SET THE FOLLOWING OF WAGO
       DD 36 00 10   LD (IX),10                          VECTOR PACKET:
       DD 36 0C 03   LD (IX+0C),03                       MR TO 10H
1A91H  DD 36 08 40   LD (IX+08),40                    Y CHECKS MASK TO 03
       DD 36 06 48   LD (IX+06),48                       ΔY TO 40H
       DD 36 0B 0A   LD (IX+0B),0A                       XH TO 48H
                                                         YH TO 0AH
```

```
          CD 50 1D        CALL ROUTINE@1D50_H
1AA0_H    18 0B           JR  }- DO NOT WRITE CACTUS IN CENTER OF SCREEN
          3E 08           LD A, 08        }- EXPAND WITH BROWN ON A
          D3 19           OUT (19_H), A   }  YELLOW BACKGROUND
          FF              RST 38_H                                   WRITE A CACTU.
          23              SUB# 34 - SUCK REQ'D                         IN THE
          4C 2A           REG. E, D }- X,Y COORDINATE              - CENTER OF THE SCRI
          08                     A  }- MR - EXPAND IT
          F4 1E                 L, H }- CACTUS PATTERN
                                        ADDRESS
          11 12 00        LD DE, 0012_H   DE = AMOUNT TO ADD TO IX           SET THE
                                               TO POINT AT NEXT V⃗P             MR
1AB0_H    DD 21 18 4F     LD IX, 4F18_H   IX = ADDRESS OF 1ST V⃗P             OF EACI
          01 20 04        LD BC, 0420_H     SET LOOP COUNTER TO 4            BULLET
                                              C = 20_H
          3E 02           LD A, 02                                           V⃗P TO
          B8              CP B        Z=0, IF B≠2                           - 20 OR 6
          20 02           JR NZ,      IF Z=0, JUMP
          0E 60           LD C, 60
          DD 71 00        LD (IX), C      SET THE MR                         SET EACH
1AC1_H    DD 36 07 01     LD (IX+07), 01          X CHECKS MASK              X CHECKS M
                                                                              TO 01,
          DD 36 0C 03     LD (IX+0C), 03          Y CHECKS MASK              Y CHECKS M
          DD 19           ADD IX, DE      POINT IX AT NEXT V⃗P                 TO 03.
          10 EC ←-20      DJNZ            LOOP BACK IF ANOTHER V⃗P TO DO
          3E 1D           LD A, 1D     }- SET PAGE OF INTERRUPT VECTO
          ED 47           LD I, A
1AD1_H    3E 74           LD A, 74     }- SET LINE
          D3 0D           OUT (0D_H), A
          FF              RST 38_H              LET'S START THE INTERRUPT'S
          51              SUB# 80 - SUCK REQ'D  STOP MAIN PROGRAM AWHILE
          64              WAIT 100_D INTERRUPTS
          F3              DI
          DD 21 0D 02     LD IX, 0D02           }- SET IX_CB TO 020D_H
                                                   FOR DISPLAYING BULLETS
          FF              RST 38_H
1ADE_H    00              BEGIN MULTIPLE CALLS
```

MAIN PROGRAM CONT'D

```
         2B      SUB#42 - SUCK REQ'D                          BLANK THE WORD:
1AE0H    12      REG. E  ]- # BYTES/LINE
         08           D  ]- # OF LINES HIGH                   "GET READY"
         FF           B  ]- BLANK WITH FFH
         33 40       L,H ]- ADDRESS AT WHICH
                            TO START BLANKING
         35      SUB #52 - SUCK REQ'D
         40,01   REG. E,D ]- X,Y COORDINATES                  DISPLAY
         0B             C ]- CHAR DISPLAY PARAMETERS          "DRAW"
         87 1D        L,H ]- STRING ADDRESS
         33      SUB#50 - SUCK REQ'D                          WRITE
         20 02   REG. E,D ]- X,Y COORDINATES                  THE FIRST
         0B             C ]- CHAR DISPLAY PARAMETERS          BULLET
         BB             A ]- CHAR CODE                        AT TOP OF SCREEN
1AF0H    07      SUB #6 - SUCK REQ'D                          WRITE THE REMAIN.
         53 1B   REG. L,H ]- NESTED MULTIPLE                  5 BULLETS AT TOP OF
                             CALLS @ 1B53H                    SCREEN.
         0D      SUB # 12 - REG. LOAD REQ'D                        SET
         01      REG. B ]- SUCK E                             X COORDINATE FOR
         68      E = X COORDINATE                             PLAYER 2 BULLET
                                                                 TO 68H
         32      SUB#50 - NO SUCK REQ'D                       DISPLAY PLAYER 2
1AF7H    07      SUB #6 - SUCK REQ'D                          BULLETS AT TOP OF
         53 1B   REG. L,H ]- NESTED MULTIPLE                  SCREEN
                             CALLS @ 1B53H
         51      SUB# 80 - SUCK REQ'D
         3C      WAIT FOR 3CH INTERRUPTS                      WAIT A BIT
         2B      SUB#42 - SUCK REQ'D                          AND BLANK
         08      REG. E  ]- # BYTES/LINE                      THE WORDS
         08           D  ]- # OF LINES HIGH                   "DRAW"
         FF           B  ]- BLANK WITH FFH
1B00H    38 40       L,H ]- ADDRESS AT WHICH
                            TO START BLANKING
         02      END MULTIPLE CALLS
1B03H    FF      RST 38H
         00      BEGIN MULTIPLE CALLS                         START
         43      SUB#66 - SUCK REQ'D                          THE
         14 02   KEYBOARD "MASK TABLE" POINTER                SENTRY
```

```
        45          SUB#68 - SUCK REQD         ] LOOK AT THE DO IT
        34 1B       DO IT TABLE ADDRESS        ]  TABLE *
        02          END MULTIPLE CALLS
        DD 21 18 4F  LD IX, 4F18H              ] IX=ADDRESS OF 1ST
                                                    BULLET V⃗P
        11 12 00    LD DE, 0012H               ] DE= AMOUNT TO ADD TO
                                                    TO POINT TO NEXT V⃗P
1B13H   06 04       LD B, 04                   ] SET LOOP COUNTER TO 4
        C5          PUSH BC                    ]     SAVE
        D5          PUSH DE                    ] THE LOOP COUNTER
                                                    (DE)
        CD CE 18    CALL ROUTINE@18CEH           CALL BULLET HIT SOMETHING ROUTIN
        D1          POP DE                     ]   GET
        C1          POP BC                     ] LOOP COUNTER
                                                  THAT V⃗P Δ
        DD 19       ADD IX, DE                   POINT TO NEXT BULLET V⃗P
        3A DE 4F    LD A, (4FDEH )             ] WAS A COWBOY JUST SHOT?
1B21H   3D   -33    DEC A                      ] IF YES, DON'T BOTHER TO LOOP,
        28 DF       JR Z,   IF Z=1             ] ie, JUMP BACK TO SENTRY TO
                            JUMP               ] REPORT IT.
        10 EF -17D  DJNZ  <- DECREMENT LOOP COUNTER. IF IT ISN'T ≠0, LOOP
                             BACK TO LOOK AT ANOTHER BULLET V⃗P.
1B26H   18 DB       JR  <------------------    JUMP BACK TO SENTRY @1B03H
           -37D
```

* HIEARCHY FOR DO IT TABLE

① TIMER REACHED ZERO

② A COWBOY WAS SHOT

③ PLAYER #1 KNOB CHANGED

④ PLAYER #2 KNOB ↓

⑤ DISPLAY THE TIMER

⑥ A CALCULATOR KEY WAS PRESSED (TIME TO SHUT DOWN)

⑦ TRIGGER #1 CHANGED

⑧ JOYSTICK #1 ↓

⑨ TRIGGER #2

⑩ JOYSTICK #2

GO BACK TO BEGINNING*

```
1B28H  02        . END MULTIPLE CALLS
       C3 0C 1A    JP 1A0CH
```

* USED BY DO IT ROUTINE @ 1B08H

GAME OVER ROUTINE

```
1B2CH  3A F8 4F    LD A,(4FF8H)        ] IS BIT 7 OF GAME STATUS BYTE (4FF8H)
       CB 7F       BIT 7,A             ] IF NOT SET, RETURN.
       C8          RET Z   IF Z=1, RET ] IF SET, CALL SUB #120 TO "QUIT".
       FF          RST 38H             ] (BIT 7 IS SET BY ROUTINE @ 1964H)
1B33H  78          SUB#120 - NO SUCK REQ'D ]
```

DO IT TABLE

```
1B34H  08    ] BIT 7 OF 4FDDH WAS SET (BY ROUTINE @ 17E1H, ie, THE TIMER REACHED (
       281B  ] FINISH "DO IT" ROUTINE, THEN JUMP TO 1B28H.

       09    ] BIT 0 OF 4FDEH WAS SET (BY ROUTINE @ 18CEH, ie, A COWBOY WAS SHOT
       281B  ] FINISH "DO IT" ROUTINE, THEN JUMP TO 1B28H

       5C    ] PLAYER #1 KNOB CHANGED
       B918  ] CALL ROUTINE @ 18B9H, THEN FINISH "DO IT" ROUTINE

       5D    ] PLAYER #2 KNOB CHANGED
       B118  ] CALL ROUTINE @ 18B1H, THEN FINISH "DO IT" ROUTINE

       55    ] PLAYER #1 JOYSTICK CHANGED
       8F18  ] CALL ROUTINE @ 188FH, THEN FINISH "DO IT" ROUTINE

       57    ] PLAYER #2 JOYSTICK CHANGED
       9518  ] CALL ROUTINE @ 1895H, THEN FINISH "DO IT" ROUTINE
```

```
1B46H  93    ]  A CALCULATOR KEY IS DOWN (TIME TO SHUT DOWN)
       8D18  ]  INITIATE A MULTIPLE SUB CALL @188DH, THEN FINISH "DO IT" ROUTINE

       54    ]  PLAYER #1 TRIGGER CHANGED
       FF17  ]  CALL ROUTINE @17FFH, THEN FINISH "DO IT" ROUTINE

       56    ]  PLAYER #2 TRIGGER CHANGED
       0A18  ]  CALL ROUTINE @180AH, THEN FINISH "DO IT" ROUTINE

       51    ]  DISPLAY THE TIMER
       E117  ]  CALL ROUTINE @17E1H, THEN FINISH "DO IT" ROUTINE

1B52H  C0       "DO IT" TABLE TERMINATOR
```

DISPLAY 5 MORE BULLETS*

```
1B53H  32   SUB#50- NO SUCK REQ'D  ]
       32         |                ]  DISPLAY THE 5
       32         |                ]  REMAINING BULLETS
       32         |                ]
       32         v                ]
1B58H  08                             ABORT THE NEST
```

* USED BY ROUTINE @1AF0H, 1AF7H

INTERRUPT ROUTINE #2

```
1B59H  08           EX AF,AF'                      ] SAVE
       D9           EXX                            ] THE
       DD E5        PUSH IX                        ] ENVIRONMENT
       3E 74        LD A,74H                       ] SET LINE OF INTERRUPT
       D3 0D        OUT (0D),A                     ] VECTOR BACK TO 74H
1B61H  3E C8        LD A,C8H                       ] SET # OF LINES EACH INTERRUPT
       D3 0F        OUT (0FH),A                    ] BACK TO C8H
       21 12 4F     LD HL,4F12H
       CD 67 1D     CALL ROUTINE @1D67H
       CD 25 1D     CALL ROUTINE @1D25H
       AF           XOR A                          ]
       32 FF 0F     LD (0FFFH),A                   ]
1B72H  DD CB 01 46  BIT 0,(IX+01)                  ] IS BIT 0 OF COWBOY VECTOR STATUS SET?
       20 28        JR NZ,     IF Z=0, JUMP        ] IF YES, JUMP
                                                     IF NO, CONTINE.
1B78H  11 05 14     LD DE,1405H                    Y SIZE, X SIZE BYTES TO BLANK
       FF           RST 38H                        ] BLANK THE
       28           SUB#40 - NOSUCK REQ'D          ] COWBOY
       26 1E        LD H,1EH                       ]
       DD 6E 12     LD L,(IX+12H)                  ] HL=ADDRESS OF COWBOY LEGS
       2C           INC L                          ]
       2C           INC L                          ]
       FF           RST 38H                        ] DISPLAY THE COWBOY'S LEGS
       1E           SUB#30 - NO SUCK REQ'D         ]
       DD CB 01 6E  BIT 5,(IX+01)                  ] IS BIT 5 OF VECTOR STATUS SET?
       20 30        JR NZ,        IF Z=0, JUMP     ] IF YES, JUMP TO 1BBCH
                                                     IF NOT, CONTINUE
       21 D7 1D     LD HL,1DD7H      HL= ADDRESS OF SHOOTING       ]
                                         ARM PATTERN ADDRESS       ]
                                         TABLE                     ]
       16 00        LD D,00H                                       ]
       DD 5E 0F     LD E,(IX+0FH)    ] GET THE INDEX               ] DISPLAY THE
       19           ADD HL,DE        ] ADD IT TO HL                ] COWBOY
       5E           LD E,(HL)        ] PUT INDEXED                 ] SHOOTING
       23           INC HL           ] PATTERN ADDRESS             ] ARM
       56           LD D,(HL)        ] IN HL                       ]
1B98H  EB           EX DE,HL         ]                             ]
```

```
1B99H  FF          RST 38H                          DISPLAY THE
       1E          SUB#30 - NO SUCK REQ'D           ARM
       21 0C 1F    LD HL, 1F0C                 HL = COWBOY PATTERN ADDRESS
       18 08       JR
       11 04 16    LD DE, 1604H                     BLANK
       FF          RST 38H                          THE
       28          SUB# 40 - NO SUCK REQ'D          WAGON
       21 3C 1F    LD HL, 1F3CH                HL = WAGON PATTERN ADDRESS
1BA8H  FF          RST 38H                          WRITE THE
       1E          SUB#30 - NO SUCK REQ'D           COWBOY/WAGON
       DD 72 0E    LD (IX+0EH), D                   SAVE THE COWBOY/WAGON
       DD 73 0D    LD (IX+0DH), E                   MAGIC BLANK ADDRESS
       21 15 4F    LD HL, 4F15H
       CD 50 1D    CALL ROUTINE @ 1D50H
       DD E1       POP IX
       08          EX AF, AF'
       D9          EXX
       FB          EI
1BBBH  C9          RET
```

```
1BBCH  21 08 1F    LD HL, 1F08H
1BBFH  18 E7       JR              JUMP TO 1BA8H
          ↑
         -25
```

INTERRUPT ROUTINE #1

```
1BC1H  F5           PUSH AF        ]
       C5            |   BC        ]
       D5            |   DE        ]- SAVE THE ENVIRONMENT
       E5            |   HL        ]
       DD E5         ↓   IX        ]
       21 19 4F     LD HL, 4F19H       HL= VECTOR STATUS ADDRESS      ]  SET THI
                                           OF 1ST BULLET V⃗P.          ]  TIME
       11 11 00     LD DE, 0011H       DE = AMOUNT TO ADD TO HL TO    ]  BASE
                                            POINT AT NEXT VECTOR STATUS]- OF EAC
       06 04        LD B, 04H          SET LOOP COUNTER TO 4          ]  BULLEI
       CD 1A1D      CALL ROUTINE @ 1D1A TO SET THE TIME BASES         ]  V⃗P
1BD2H  23           INC HL           } ADJUST HL TO POINT AT VECTOR   ]  SET TH
                                       STATUS OF 1ST PLAYER V⃗P        ]  TIME
       11 16 00     LD DE, 0016H       DE = AMOUNT TO ADD TO HL TO    ]  BASE
                                            POINT AT NEXT VECTOR STATUS]- OF EAC
       06 03        LD B, 03           SET LOOP COUNTER TO 3          ]  COWBOY
       CD 1A1D      CALL ROUTINE @ 1D1A TO SET THE TIME BASES         ]  WAGON V⃗
       AF           XOR A            ]
       32 FF0F      LD (0FFF), A     ]- ZERO 4FFFH ?
       06 04        LD B, 04           SET LOOP COUNTER TO 4                          ]
1BE1H  DD 21 18 4F  LD IX, 4F18H       IX= ADDRESS OF 1ST                             ]
                                           BULLET VECTOR PACKET                       ]
       DD CB 01 76  BIT 6, (IX+01)   ] IS VECTOR STATUS BLANK BIT SET?                ]
       28 11        JR Z,  IF Z=1,   ]- IF NO, JUMP (NO BLANKING REQ'D)               ]
                           JUMP      ] IF YES, CONTINUE.                              ]  BLANI
       DD 66 0E     LD H, (IX+0E)    ]- GET THE BULLETS MAGIC                         ]  AND/o
       DD 6E 0D     LD L, (IX+0D)    ]    BLANK ADDRESS                               ]  WRITE
1BF1H  DD 7E 0F     LD A, (IX+0F)    ]  OUTPUT THE MAGIC REGISTER                     ]  THE
       D3 0C        OUT (0C), A      ]- WITH ITS OLD VALUE                            ]- BULLET
       36 C0        LD (HL), C0      }- BLANK THE OLD BULLET                          ]  IF
       DD CB 01 B6  RES 6, (IX+01)   }- TURN OFF BLANK BIT                            ]  REQ'D
       DD CB 01 7E  BIT 7, (IX+01)   ] IS VECTOR STATUS ACTIVE BIT SET?               ]
1C00H  28 2B        JR Z,  IF Z=1,   ]- IF NO, JUMP (NO BULLET WRITE REQ'D)           ]
                           JUMP      ] IF YES, CONTINUE.                              ]
       DD 56 0B     LD D, (IX+0B)    ]  GET X,Y        ]                              ]  (LOOK A
       DD 5E 06     LD E, (IX+06)    ]- COORDINATES    ]  DETERMINE                   ]  ALL 4
       DD 7E 00     LD A, (IX)       }- GET MR         ]- NEW                         ]  BULLET
       FF           RST 38H                            ]  MAGIC ADDRESS               ]  V⃗P'S
       38           SUB#56 - NO SUCK                   ]                              ]
                             REQ'D                                                    ]
       DD 72 0E     LD (IX+0E), D    ]     SAVE NEW                                   ]
1C10H  DD 73 0D     LD (IX+0D), E    ]- MAGIC ADDRESS                                 ]
                                     ]  FOR BLANKING                                  ]
       DD 77 0F      LD (IX+0F), A             SAVE NEW MAGIC REGISTER
                                                 VALUE FOR BLANKING
       21 00 40      LD HL, 4000      ]
       19            ADD HL, DE       ]- HL = NEW SCREEN ADDRESS
       7E            LD A, (HL)               A = CONTENTS OF THAT
                                                SCREEN ADDRESS
       EB            EX DE, HL                DE = NEW SCREEN ADDRESS
                                              HL = NEW MAGIC ADDRESS
       36 C0         LD (HL), C0              WRITE THE NEW BULLET
       B7            OR A        Z=1,       ] WAS THERE SOMETHING ALREADY
                                 IF A=00    - WRITTEN AT THAT NEW SCREEN
       28 08         JR Z,       IF Z=1,      ADDRESS ?
                                 JUMP       ] IF YES, CONTINUE. IF NO, JUMP
1C21H  DD CB 01 BE   RES 7, (IX+01)   ]  KILL THE ACTIVE BIT
       DD CB 01 EE   SET 5, (IX+01)   [  SET THE
       DD CB 01 F6   SET 6, (IX+01)   ]  SET THE BLANK BIT
       11 12 00      LD DE, 0012H     ]
1C30H  DD 19         ADD IX, DE       ]- HL = ADDRESS OF NEXT
                                              VECTOR PACKET
       10 B1 -79     DJNZ      B=B-1  }  IS THERE ANOTHER BULLET
                                         VP TO LOOK AT? LOOP BACK IF YES
       21 12 4F      LD HL, 4F12
       CD 67 1D      CALL
       28 12         JR Z,      IF Z=1, JUMP
       3E 76         LD A, 76         ]
       D3 0D         OUT (0D), A      ]- SET LINE OF INTERRUPT VECTOR
                                             NOW TO 76H
1C40H  DD 7E 0B      LD A, (IX+0B)    ]                 GET
       FE 32         CP 32H  C=0, IF A≥32H     IF IT IS ≥32H, GENERATE
                                               AN INTERRUPT EVERY ≈0 SEC
       3E 00         LD A, 00H                 IF IT IS <32H, GENERATE
       30 02         JR NC,   IF C=0, JUMP     AN INTERRUPT REQUEST
       3E 6A         LD A, 6A                  EVERY 106D LINES (.007 SEC
       D3 0F         OUT (0FH), A     ]
       FB            EI
       DD 21 18 4F   LD IX, 4F18H             IX = ADDRESS OF 1ST
                                                   BULLET VECTOR PACKET
1C52H  06 04         LD B, 04H                SET LOOP COUNTER TO 4
       21 8B 1D      LD HL, 1D8BH             HL = LIMITS TABLE ADDRESS
       11 12 00      LD DE, 0012H             DE = AMOUNT TO ADD TO IX
                                                   TO POINT AT NEXT VP
       DD CB 01 7E   BIT 7, (IX+01)   ]  IS ACTIVE BIT SET?
                               IF Z=1, -  IF NOT, DON'T UPDATE VP
       28 0C         JR Z,     JUMP   ]  IF YES, UPDATE VP
1C60H  FF            RST 38H          ]  UPDATE THE VECTOR
       3E            SUB#62 - NO SUCK -  PACKET. ROUTINE WILL
                              REQ'D   ]  ABORT IF TIME
                                         BASE = 0.
```

INTERRUPT ROUTINE #1 CONT'D

```
        DD CB 07 5E   BIT 3, (IX+07)  ]  IF UPDATED X HAS               UPDATE
        28 04         JR Z,  IF Z=1,  ]- REACHED A LIMIT, KILL          THE BU
                             JUMP     ]  THE VECTOR ACTIVE BIT.       - X,Y
        DD CB 01 BE   RES 7, (IX+01)  ]                                 COORDINA
        DD 19         ADD IX, DE         IX = ADDRESS OF NEXT            IF REQ'
                                              BULLET VECTOR PACKET
        10 EA         DJNZ     B=B-1 }  LOOP BACK IT THERE IS
                                        ANOTHER BULLET VP To UPDATE
1C70H   06 02         LD B, 02       }  SET LOOP COUNTER TO 2.
        21 15 4F      LD HL, 4F15H
        CD 67 1D      CALL
        CA F8 1C      JP Z, 1CF8H    IF Z=1,
                                     JUMP
        CD 25 1D      CALL
        FB            EI
        DD CB 01 46   BIT 0, (IX+01)  ]- IS BIT 1 OF VECTOR STATUS SE
1C83H   C2 03 1D      JPNZ, 1D03H  IF Z=0,
                                   JUMP
        DD CB 01 6E   BIT 5, (IX+01)  ]- IS BIT 5 OF VECTOR STATUS SET
        20 25         JR NZ,
        DD 7E 03      LD A, (IX+03)   ]
        DD B6 04      OR (IX+04)      ]  ARE BOTH ΔX AND ΔY=0
1C92H   DD B6 08      OR (IX+08)      ]-
        DD B6 09      OR (IX+09)      ]
        20 17         JR NZ,  IF Z=0,
                              JUMP
        DD 77 02      LD (IX+02), A      ZERO TIME BASE
        DD CB 01 66   BIT 4, (IX+01)  ]- IS BIT 4 OF VECTOR STATUS SE
1CA1H   20 36         JR NZ,  IF Z=0
                              JUMP
        DD 36 12 4B   LD (IX+12), 4B
        DD CB 01 DE   SET 3, (IX+01)  ]- SET BITS 3,4 OF VECTOR STAT
        DD CB 01 E6   SET 4, (IX+01)  ]
        18 28         JR
1CB1H   21 83 1D      LD HL, 1D83H    HL = LIMITS TABLE
                                           ADDRESS
        FF            RST 38H          ]- UPDATE THE
        3E            SUB# 62 - NO SUCK]  VECTOR PACKET
                                REQ'D
        28 08         JR Z,
        DD CB 01 DE   SET 3, (IX+01)
        DD CB 01 A6   RES 4, (IX+01)
1CC0H   DD 7E 11      LD A, (IX+11H)
```

```
          91           SUB C
          F2 D6 1C     JP P, 1CD6H
          DD 5E 12     LD E, (IX+12H) ]          LAST
          16 1E        LD D, 1EH      ]- DE = COWBOY FOOT
                                              PATTERN ADDRESS-2
          1A           LD A, (DE)     ]  GET THE NEXT LOW ORDER
          DD 77 12     LD (IX+12H), A ]- ADDRESS-2 OF COWBOY
                                         FOOT PATTERN AND SAVE IT
1CD0H     13           INC DE         ]
          1A           LD A, (DE)     ]-      A=04
          DD CB 01 DE  SET 3, (IX+01) ]- SET BIT 3 OF
                                         VECTOR STATUS
          DD 77 11     LD (IX+11H), A ]-  (IX+11H) <- 04
          DD 7E 0F     LD A, (IX+0F)   ]  IS (IX+10H) = SHOOTING ARM
                                                        PATTERN ADDRESS INDEX
          DD BE 10     CP (IX+10H)     ]- IF NOT, SKIP NEXT 2 INSTRUCTIONS.
          28 07        JR Z,  IF Z=1,  ]  IF YES, CONTINUE
                              JUMP
1CE1H     DD CB 01 DE  SET 3, (IX+01)  ]- SET BIT 3 OF
                                          VECTOR STATUS
          DD 77 10     LD (IX+10H), A  ]- (IX+10H) <- SHOOTING ARM
                                                      PATTERN ADDRESS INDEX
          DD CB 01 5E  BIT 3, (IX+01)  ]- IS BIT 3 OF VECTOR STATUS SET?
          20 20        JR NZ, IF Z=0,  ]  IF YES, JUMP TO 1D0EH .
                              JUMP
          21 15 4F     LD HL, 4F15H
1CF1H     CD 50 1D     CALL
          05           DEC B            ]  DECREMENT LOOP COUNTER.
          C2 75 1C     JP NZ,  IF Z=0,  ]- IF B=1, LOOP BACK TO LOOK AT 2ND
                               JUMP     ]           COWBOY VP @ 1C75H.
                                           IF B=0, CONTINUE
1CF8H     FB           EI               ]-
          CD 00 02     CALL  ]- DO THE ON-BOARD TIMER ROUTINE
          DD E1        POP  IX  ]
          E1            |   HL  ]      RESTORE
          D1            |   DE  ]-       THE
1D00H     C1            |   BC  ]      ENVIRONMENT
          F1            v   AF  ]
1D02H     C9           RET
```

```
1D03H 21 78 1D     LD HL, 1D78H                    HL = WAGON LIMITS TABLE ADDRE
      FF           RST 38H                         ]
      3E           SUB # 62 -  NO SUCK             ]- UPDATE WAGON VECTOR
                               REQ'D               ]
      21 15 4F     LD HL, 4F15H
      CD 25 1D     CALL ROUTINE @ 1D25H
      DD CB 01 9E  RES 3, (IX+01)
      21 12 4F     LD HL, 4F12H
      CD 50 1D     CALL ROUTINE @ 1D50H
1D18  18 DE        JR                              JUMP TO LOCATION 1CF8H
```

SET TIME BASE

```
1D1AH 7E           LD A, (HL)                          PUT VECTOR STATUS IN A
      23           INC HL                              POINT HL AT TIME BASE
      E6 A0        AND A0H     Z=1,                 ]  IS ONLY VECTOR STATUS BIT. AND BLANI
                               IF A=A0H             ]  BIT SET?
      28 01        JR Z,       IF Z=1, JUMP         ]- IF NO, INCREMENT TIME BASE.
      34           INC (HL)                         ]  IF YES, LEAVE TIME BASE ALONE.
      19           ADD HL, DE                      -POINT HL AT NEXT VECTOR STATUS
      10 F6 ← -10D DJNZ        B=B-1  IF B≠0, LOOP  ]
                                             BACK   ]- RETURN WHEN LAST TIME BASE
                                      IF B=0, RET   ]  IS SET
1D24H C9           RET                              ]
```

```
1D25H F3           DI
      DD 7E FF     LD A, (IX+FF)   ← -1
      77           LD (HL), A
      A7           AND A           Z=1, IF A=0
      C0           RET NZ          IF Z=0, RET
      23           INC HL
      77           LD (HL), A
      2B           DEC HL
1D2FH C9           RET
```

```
1D30H  DD 36 03 32    LD (IX+03), 32        ]  INITIALIZE COWBOY:
                                            ]    ΔXL TO 32H
       DD 36 01 80    LD (IX+01), 80        ]  VECTOR STATUS TO 80H
       DD 36 07 01    LD (IX+07), 01        ]  X CHECKS MASK TO 01H
       DD 36 0C 01    LD (IX+0C), 01        ]- Y      ↓       TO 01H
       DD 36 06 04    LD (IX+06), 04        ]       XH TO 04H
       DD 36 0B 28    LD (IX+0B), 28        ]       YH TO 28H
       DD 36 0F 06    LD (IX+0F), 06        ]  SHOOTING ARM PATTERN
                                            ]   ADDRESS INDEX TO 06H
       DD 36 12 4B    LD (IX+12H), 4B       ]  LOW ORDER ADDRESS-2 OF
                                            ]    COWBOY LEGS PATTERN
1D50H  DD E5          PUSH IX               ]- PUT IX (V⃗P POINTER) IN DE
       D1             POP DE                ]
       F3             DI
                             -01
       DD 36 FF 00    LD(IX+FF), 00
       23             INC HL
       7E             LD A, (HL)
       73             LD (HL), E                SAYE THE LOW ORDER ADDRESS
                                                 OF THE V⃗P POINTER
       A7             AND A   Z=1, IF A=0        ]  ARE WE INITIALIZING ?
       28 06          JR Z,       IF Z=1, JUMP   ]- IF SO, JUMP.
       5F             LD E, A
       7E             LD A, (HL)
       2B             DEC HL
       1B             DEC DE
       12             LD (DE), A
       C9             RET
       2B             DEC HL
       73             LD (HL), E
1D66H  C9             RET
```

```
1D67H  F3             DI
       5E             LD E, (HL)
       23             INC HL
       23             INC HL
       56             LD D, (HL)
       2B             DEC HL
       2B             DEC HL
       7B             LD A, E
       A7             AND A      Z=1, IF A=0
       D5             PUSH DE
       DD E1          POP IX
1D72   C9             RET
```

```
1D74    C1 1B   VECTOR #1        ]  IM2
1D76    59 1B     ↓     #2       ]  VECTOR TABLE
1D78    0A    Y LOWER LIMIT      ]  WAGON
1D79    44    Y UPPER LIMIT      ]  LIMITS TABLE
1D7A_H  47 45 54 20 52 45 41 44 59
        G  E  T     R  E  A  D  Y
1D83_H  00    LOWER X  LIMIT     ]
        2F    UPPER X    |       ]  COWBOY
        0A    LOWER Y    |       ]  LIMITS TABLE
        48    UPPER Y    ↓       ]
1D87_H  44 52 41 57
        D  R  A  W
1D8B_H  00    LOWER X  LIMIT     ]
        9F    UPPER X    |       ]  BULLETS
        09    LOWER Y    |       ]  LIMITS TABLE
        5B    UPPER Y    ↓       ]
1D8F_H  00    BULLET  ΔX_L           ]               ]
        03      |     ΔX_H           ]               ]
        0F      |     RELATIVE X_H   ]  FRAME 0      ]  BULLET
        00      |     ΔY_L           ]               ]  VECTOR PACKET
        03      |     ΔY_H           ]               ]  INITIALIZATION
        0F      ↓     RELATIVE Y_H   ]               ]  FOR RELATIVE
        00 04 0F 00 02 0C            FRAME 1         ]  SHOOTING ARM
        00 04 0F 00 01 0B              |     2       ]
        00 04 0F 00 00 08              |     3       ]
        00 04 0F 00 FF 06              |     4       ]
        00 04 0F 00 FE 04              |     5       ]
        00 03 0F 00 FD 03              ↓     6       ]
1DB9_H  48    COWBOY #2  BOTTOM  CACTUS  Y  COORDINATE
        16       |       TOP       ↓         |
        2C       |       MIDDLE              |
        43       |       BOTTOM  TREE        |
        0E       ↓       TOP     TREE        ↓
1DBE_H  12    COWBOY #1  TOP     CACTUS  Y  COORDINATE
        44       |       BOTTOM          |
        28       ↓       MIDDLE          ↓
```

1DC1$_H$	0D	PLAYER #1	TOP	TREE Y COORDINATE
	3F		BOTTOM	

1DC3$_H$	A2	COLOR REG	7	PIXEL	11	GREEN	COLOR SPECS TABLE
	7D		6		10	BROWN	
	0B		5		01	BLUE	
	87		4		00	YELLOW	
	A2		3		11	GREEN	
	7D		2		10	BROWN	
	6C		1		01	RED	
	87		0		00	YELLOW	

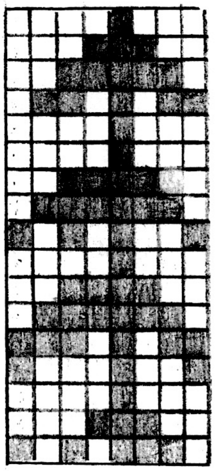
1DCB$_H$	06	# OF COWBOY 1 BULLETS LEFT	LOADED INTO 4FDA*
	06	# OF COWBOY 2 BULLETS LEFT	4FDB
	00		4FDC
	00		4FDD
	00		4FDE
	30	COWBOY 1 LAST KNOB INPUT VALUE	4FDF
	30	COWBOY 2 LAST KNOB INPUT VALUE	4FE0
	00		4FE1
	80	LAST CALCULATOR INPUT / 1PPS MARKER	4FE2
	0F	COWBOY 1 LAST JOYSTICK / TRIGGER STATUS	4FE3
	0F	COWBOY 2 LAST JOYSTICK / TRIGGER STATUS	4FE4$_H$

* INITIAL VALUES

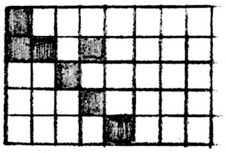
1DD7	F8 1D	FRAME	0	COWBOY SHOOTING ARM PATTERN ADDRESS TABLE
	06 1E		1	
	10 1E		2	
	18 1E		3	
	24 1E		4	
	32 1E		5	
	42 1E		6	

1DE5$_H$	01	X	SIZE	(# OF BYTES WIDE)
	11	Y		(# OF LINES HIGH)
	08	LINE	1	
	1C		2	
	3E		3	
	6B		4	

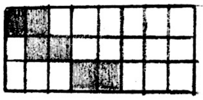
```
1DEB_H   08     LINE      5
         08      |        6
         3C      |        7
         7E      |        8
         A9      |        9
         08      |        A
         3C      |        B
         7E      |        C
         EB      |        D
         89      |        E
         08      |        F
         1C      |        10
         AE      ↓        11
1DF8_H   0A     RELATIVE  X
         0A      ↓        Y
         02     X SIZE
         05     Y SIZE
         40 00    LINE 1
         51 00     |   2
         04 00     |   3
         01 00     |   4
         00 40     ↓   5
1E06_H   0A     RELATIVE  X
         0A      ↓        Y
         02     X SIZE
         03     Y SIZE
         50 00    LINE 1
         14 00     |   2
         01 40     ↓   3
1E10_H   0A     RELATIVE  X
         0A      ↓        Y
         02     X SIZE
         02     Y SIZE
         54 00    LINE 1
         55 40     ↓   2
```



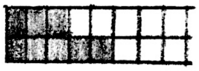
TREE (SHOWN EXPANDED)



COWBOY SHOOTING ARM (FRAME 0)

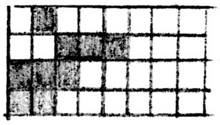


COWBOY SHOOTING ARM (FRAME 1)



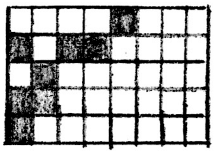
COWBOY SHOOTING ARM (FRAME 2)

```
1E18H  0A     RELATIVE X
       07        ↓     Y
       02     X SIZE
       04     Y SIZE
       10 00   LINE 1
       05 40    |   2
       54 00    |   3
       50 00    ↓   4
```



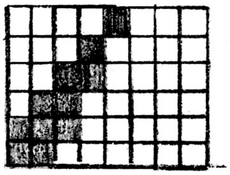
COWBOY
SHOOTING
ARM
(FRAME 3)

```
1E24H  0A     RELATIVE X
       06        ↓     Y
       02     X SIZE
       05     Y SIZE
       00 40   LINE 1
       45 00    |   2
       10 00    |   3
       50 00    ↓   4
```



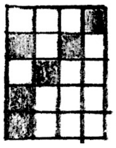
COWBOY
SHOOTING
ARM
(FRAME 4)

```
1E32H  0A     RELATIVE X
       05        ↓     Y
       02     X SIZE
       06     Y SIZE
       00 40   LINE 1
       01 00    |   2
       05 00    |   3
       14 00    |   4
       54 00    |   5
       50 00    ↓   6
```



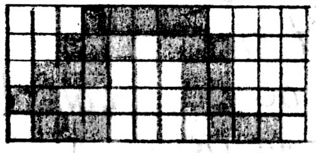
COWBOY
SHOOTING
ARM
(FRAME 5)

```
1E42H  0A     RELATIVE X
       05        ↓     Y
       01     X SIZE
       05     Y SIZE
       01     LINE 1
       44      |   2
       10      |   3
       40      |   4
       40      ↓   5
```



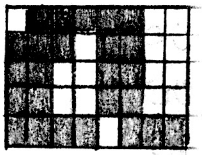
COWBOY
SHOOTING
ARM
(FRAME 6)

```
1E4B   60  *
1E4C   04
1E4DH  00   RELATIVE X
       0F      ↓     Y
       03     X SIZE
       05     Y SIZE
       01 55 00    LINE 1
       05 45 40     |   2
       15 01 40     |   3
       50 01 40     |   4
       15 00 54     ↓   5
```



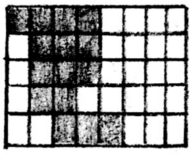
COWBOY LEGS (FRAME 0)

```
1E60   70  *
1E61   04
1E62H  02  RELATIVE X
       0F     ↓     Y
       02    X SIZE
       05    Y  ↓
       15 50    LINE 1
       54 50     |   2
       50 50     |   3
       50 50     |   4
       55 15     ↓   5
```



COWBOY LEGS (FRAME 1)

```
1E70   4B  *
1E71   04
1E72H  03  RELATIVE X
       0F     ↓     Y
       02    X SIZE
       05    Y SIZE
       55 00
       15 00
       15 00
       14 00
       05 40
```

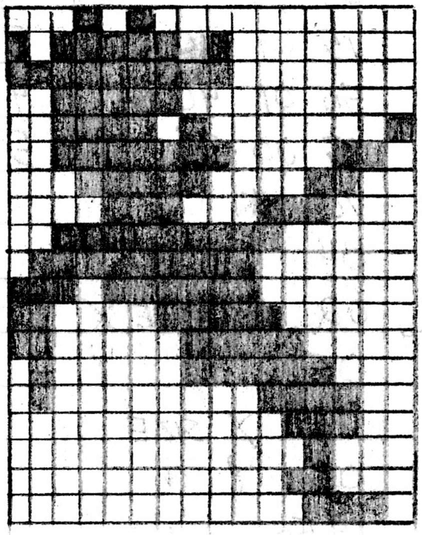


COWBOY LEGS (FRAME 2)

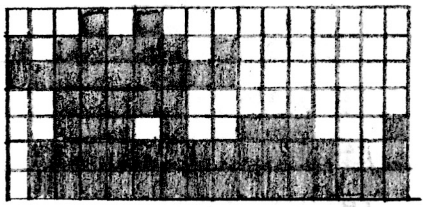
* LOW ORDER ADDRESS OF NEXT LEGS "PATTERN ADDRESS -2" TO USE

** LOW ORDER OF NEXT "FALL" PATTERN TO USE

```
1E80   D2 **
1E81   14
1E82H  00  RELATIVE X
       01     ↓     Y
       04   X SIZE
       13   Y SIZE
       01 10 00 00   LINE 1
       45 54 40 00    |   2
       55 55 40 00    |   3
       0A A8 00 00    |   4
       0A A2 00 01    |   5
       0A AA 80 14    |   6
       02 AA 00 50    |   7
       00 A8 05 40    |   8
       05 55 54 00    |   9
       15 55 50 00    |   A
       54 55 50 00    |   B
       50 05 54 00    |   C
       50 01 55 00    |   D
       10 01 55 40    |   E
       10 00 05 50    |   F
       00 00 01 50    |   10
       00 00 00 40    |   11
       00 00 01 40    |   12
       00 00 00 54    ↓   13
1ED2H  D2 **
       3C
       00  RELATIVE X
       0D     ↓     Y
       04   X SIZE
       07   Y SIZE
       01 10 00 00   LINE 1
       45 54 40 00    |   2
       55 55 40 00    ↓   3
```

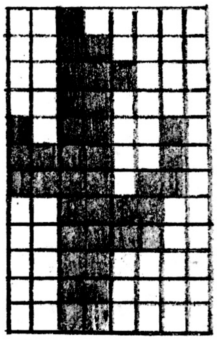


COWBOY FALLING DOWN (FRAME 0)



COWBOY FALLING DOWN (FRAME 1)

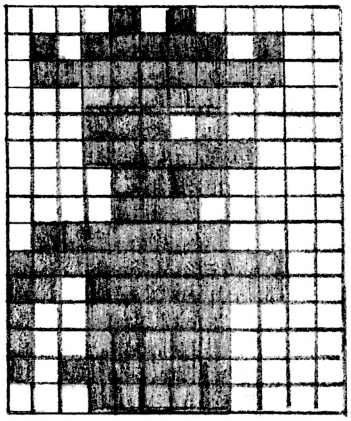
```
1EE4H  0A A8 00 00     LINE 4
       0A 88 15 01      |   5
       16 A5 55 41      |   6
       15 55 55 55      ↓   7
1EF4H  01  X SIZE
       0C  Y SIZE
       20   LINE 1
       30    |    2
       38    |    3
       30    |    4
       B2    |    5
       F2    |    6
       F6    |    7
       3C    |    8
       3C    |    9
       30    |    A
       30    |    B
       30    ↓    C
1F02H  47 4F 54 20 4D 45
       G  O  T     M  E
```



CACTUS
(SHOWN EXPANDED

```
1F08H  00 RELATIVE X
       00    ↓     Y
       01  X SIZE
       01  Y SIZE
1F0CH  00  LINE 1 (RELATIVE X)
       00  RELATIVE Y
       03   X SIZE
       0F   Y SIZE
       00 44 00  LINE 1
       11 55 10   ↓   2
```

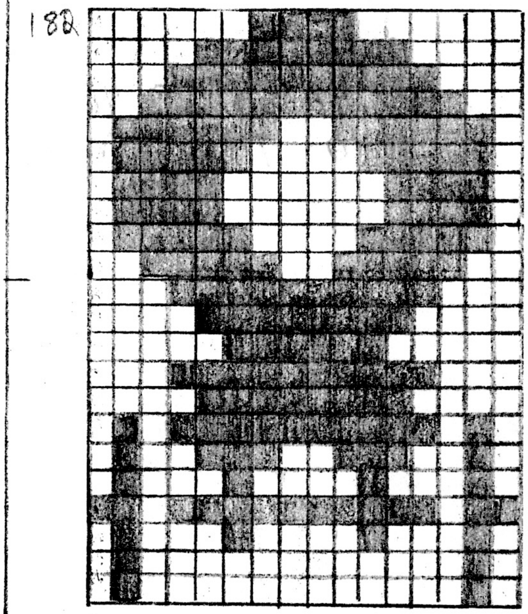
```
1F16H   15 55 50     LINE 3
        02 AA 00          4
        02 A2 00          5
        02 AA 80          6
        00 AA 00          7
        00 A8 00          8
        15 55 00          9
        55 55 50          A
        51 55 50          B
        41 55 00          C
        41 55 00          D
        45 55 00          E
        01 55 00          F
```



COWBOY *

* MINUS SHOOTING ARM AND LEGS

```
1F3DH   00  RELATIVE Y
        04  X SIZE
        16  Y SIZE
        00 05 50 00   LINE 1
        00 55 55 00        2
        01 55 55 40        3
        05 55 55 50        4
        15 54 15 54        5
        15 50 05 54        6
        15 40 01 54        7
        15 40 01 54        8
        15 50 05 54        9
        05 54 15 50        A
        01 55 55 40        B
        00 55 55 00        C
        00 15 54 00        D
        02 AA AA 80        E
        00 AA AA 00        F
        12 AA AA 84       10
        10 A8 2A 04       11
        10 20 08 04       12
```



WAGON

```
1F88H  52 AA AA 85   LINE 13
       10 20 08 04     |   14
       10 00 00 04     |   15
       10 00 00 04     ↓   16
```

Gunfight
An Instruction by Instruction Breakdown
An MCM Design Project
Documentation written by Michael Matte

Errata Sheet

This errata sheet is based on information that Michael Matte wrote in an email on May 18, 2017. In part it said:

I found 3 errors on my GUNFIGHT breakdown. Here are the errors in my GUNFIGHT breakdown:

Error 1, Page 35:

There is one byte 00H missing at 1DD3H.

```
1DD0H 30     COWBOY 1 Last Knob Input Value
      30 COWBOY 1 Last Knob Input Value
      00
1DD3H 00                                    <-- Missing byte
1DD4H 80 Last Calculator Input /1PPS Marker
```

Error 2, Page 37

Line 5 of frame 4 is missing at 1E30H.

```
1E28H 00 40  ; Line 1, frame 4
      45 00  ; Line 2, frame 4
      10 00  ; Line 3, frame 4
      50 00  ; Line 4, frame 4              <-- Missing 2 bytes
1E30H 40 00  ; Line 5, frame 4

1E32H 0A Relative X
```

Error 3, Page 42

After the wagon pattern, the sound bytes are missing beginning at 1F98H.

```
1F98H 00
```

Set Audio

```
1F99H 80 11 B0 09 00 C9
```

Home on the Range

```
1F9FH CD 99 1F 24 7E 0C 8D 12
1FA7H 96 06 A8 24 96 F0
```

Taps

```
1FADH CD 99 1F 12 BD 06 BD 24
1FB5H 8D 12 BD 06 8D 24 70 F0
```

Funeral

```
1FBDH CD 99 1F 18 E1 12 E1 06
1FC5H E1 18 E1 12 BD 06 C8 12
1FCDH C8 06 E1 12 E1 06 EE 12
1FD5H E1 F0
```

Gunshot

```
1FD7H 88 EF FF 3F 00 FF FD F5
      F0 E0 B0 FF 3F E1 05 05
      8F 05 4C F0
```

End of Gunfight Program