

WPC Sound Calls

Document History

April 10, 2012, initial draft. Basics on the sound calls. This is a compilation of various info already provided in various posts in the forum.

Document Scope

This document provides some basics on WPC sound calls. This will show you how to find the sound call table in your WPC ROM, thus allowing you to freely modify sound calls. In the past this has been deemed necessary mainly for removing offensive sound calls.

This document will also show how you might be able to find the list of sounds used during attract mode when you hit the flipper buttons.

Additionally, this document will show you how you can use the existing “Family Mode” setting, if present in your game, to determine if your game should play one sound instead of another.

At this time, this document doesn’t discuss how the sound calls get put into a queue which eventually gets pushed out to the sound card.

Disclaimer

This information is for educational and entertainment purposes only. Some of the interpretation about code may be incorrect so take some of what is presented with a grain of salt. In fact, the code comments shown in this document have been made over the course of several years. Some parts of the code were commented poorly as little was known about the code being annotated. I try to clean this up as I go.

Exercise caution when modifying ROM images as they could have real physical effects which may be undesirable, especially if you modify code that causes hardware components to operate outside of their specifications.

Finding The SoundCallTable[] (IJ_L7)

Below are 3 functions from IJ_L7 which are located in non-banked ROM.

- “DoSoundTableIndexA()” will play an 8-bit sound index from the A register.
- “DoSoundTableIndexY()” will play a 16-bit sound index from the Y register.
- “DoSoundTableIndexYStackSet()” is a common function called by both of the above functions.

For the purposes of this write-up the details of what these functions actually accomplish are not important, however you’re welcome to trace through and see my comments to get some idea what’s going on. The purpose of showing these function here are simply for the purpose of finding the location of the SoundCallTable[]. The function “DoSoundTableIndexYStackSet()” loads a pointer to the SoundCallTable[] which is the primary focus of this discussion.

In the case of IJ_L7 the SoundCallTable[] pointer is 0x81F8. By finding similar code in your ROM you should be able to find your SoundCallTable[] pointer.

```

;
;-----;-----
;
;
; DoSoundTableIndexA()
;
; A has SoundCallTable[] index to play
;
BF1D: 34 76      PSHS  U,Y,X,B,A      ; Save registers
BF1F: 32 7D      LEAS  $FFFD,S        ; Decrement stack pointer by 3
BF21: 6F 61      CLR   $0001,S        ; Clear memory on stack + 1
BF23: A7 62      STA   $0002,S        ; Save A (sound index) on stack + 2
BF25: 10 AE 61   LDY   $0001,S        ; Y gets 0x00yy, where yy is sound number to play.
BF28: 20 07      BRA   $BF31          ; goto DoSoundTableIndexYStackSet()
;
;-----;-----
;
;
; DoSoundTableIndexY()
;
BF2A: 34 76      PSHS  U,Y,X,B,A      ; Save registers
BF2C: 32 7D      LEAS  $FFFD,S        ; Decrement stack pointer by 3
BF2E: 10 AF 61   STY   $0001,S        ; Save sound index on stack + 2
```

```

; goto DoSoundTableIndexYStackSet()
;
;-----
;
; DoSoundTableIndexYStackSet ()
;
;
BF31: D6 11      LDB  $11
BF33: E7 E4      STB  ,S
BF35: 8E 81 F8   LDX  #81F8
BF38: BD AD 21   JSR  $AD21
BF3B: BD 8F FB   JSR  $8FFB
BF3E: EC 04      LDD  $0004,X
;
BF40: 85 04      BITA #04
BF42: 27 07      BEQ  $BF4B
BF44: CE 04 01   LDU  #$0401
BF47: 8D 71      BSR  $BFBA
BF49: 25 64      BCS  $BFAF
;
; if (SoundTableEntry[Y].byte5 & 0x04)
; {
;     U = 0x0401
;     Call $BFBA
;     If C-bit set, goto: End of function cleanup, C-bit Set
; }
;
; if (SoundTableEntry[Y].byte6 & 0x08)
; {
;     U = 0x03F5
;     Call $BFBA
;     If C-bit set, goto: End of function cleanup, C-bit Set
; }
;
; if (!(SoundTableEntry[Y].byte6 & 0x10))
; {
;     B &= 0x18 // keep only the bits we just examined
;     A &= 0x04 // keep only the bit we just examined
;     if (if SoundTableEntry[Y].bytes5_and_6 has any 0x0418 bit set)
;         goto $BF6E
; }
;
BF64: CE 03 E9   LDU  #$03E9
BF67: 8D 51      BSR  $BFBA
BF69: 25 44      BCS  $BFAF
BF6B: BD C0 67   JSR  $C067
;
BF6E: EC 04      LDD  $0004,X
BF70: 85 04      BITA #04
BF72: 27 09      BEQ  $BF7D
; if (SoundTableEntry[Y].byte5 & 0x04)
; {

```

BF74: CE 04 01	LDU	#\$0401	; U = 0x0401
BF77: BD C0 67	JSR	\$C067	; CheckRAM_U_andTakeAction_ClearOffsets02_03_05_06_08_09()
BF7A: BD C0 2F	JSR	\$C02F	; Check03FD_03F1_for_04bit_AmdClearOffsets()
			; }
			; }
BF7D: C5 10	BITB	#\$10	; if (SoundTableEntry[Y].byte6 & 0x10)
BF7F: 27 03	BEQ	\$BF84	; {
BF81: CE 03 E9	LDU	#\$03E9	; U=0x03E9
			; }
			; }
BF84: C5 08	BITB	#\$08	; if (SoundTableEntry[Y].byte6 & 0x08)
BF86: 27 06	BEQ	\$BF8E	; {
BF88: CE 03 F5	LDU	#\$03F5	; U=0x03F5
BF8B: BD C0 67	JSR	\$C067	; CheckRAM_U_andTakeAction_ClearOffsets02_03_05_06_08_09()
			; }
BF8E: 10 AE 61	LDY	\$0001,S	; Get byte off stack
BF91: 10 AF 4A	STY	\$000A,U	; Push it into &U+A
BF94: C5 01	BITB	#\$01	; if (!(SoundTableEntry[Y].byte6 & 0x01))
BF96: 26 05	BNE	\$BF9D	; {
			; -----
			; The SoundTableEntry[Y].byte6 0x01 bit is NOT set, this means the first
			; two bytes of the sound table are actual sound number to play.
			; -----
BF98: BD C0 95	JSR	\$C095	; play a sound?
BF9B: 20 09	BRA	\$BFA6	; Goto: End of function cleanup, C-bit Clear
			; }
			;
			; -----
			; The 0x01 bit was set in SoundTableEntry[Y].byte6, this means
			; that the first 2 bytes of sound table are pointer to a
			; series of commands to play sounds, call function to process...
			; -----
BF9D: BD C0 BC	JSR	\$C0BC	; LoadUPointerWithSoundTableEntryValues()
BFA0: BD 89 48	JSR	\$8948	; CallBankedFunction_Param_WPCAddr()
BFA3: 57 8F 3D			; -->ProcessSoundCallTableEntry()
			;
			; -----
			; End of function cleanup, C-bit Clear
			; -----
BFA6: E6 E4	LDB	, S	;
BFA8: BD 8F FB	JSR	\$8FFB	; SetPageBANK1()
BFAB: 1C FE	ANDCC	#\$00FE	;
BFAD: 20 07	BRA	\$BFB6	;
			; -----
			; End of function cleanup, C-bit Set

```

;-----
BFAF: E6 E4      LDB      ,S      ;
BFB1: BD 8F FB      JSR      $8FFB      ; SetPageBANK1(), restore original ROM bank
BFB4: 1A 01      ORCC     #$0001      ;
BFB6: 32 63      LEAS     $0003,S      ;
BFB8: 35 F6      PULS     A,B,X,Y,U,PC ; Done, RTS
;
;-----
;

```

Finding The SoundCallTable[] (TS_LH6)

To show another example, the other day I wanted to find the SoundCallTable[] in The Shadow LH-6. By searching for different byte patterns shown in the IJ_L7 (above) I quickly was able to find these series of bytes which appear to be consistent with the functions in IJ_L7:

```

;
;-----;
;
;
; DoSoundTableIndexA()
;
; A has SoundCallTable[] index to play
;
BE8C: 34 76      PSHS  U,Y,X,B,A
BE8E: 32 7D      LEAS  $FFFD,S
BE90: 6F 61      CLR   $0001,S
BE92: A7 62      STA   $0002,S
BE94: 10 AE 61   LDY   $0001,S
BE97: 20 07      BRA   $BEA0
;
;-----;
;
; DoSoundTableIndexY()
;
BE99: 34 76      PSHS  U,Y,X,B,A
BE9B: 32 7D      LEAS  $FFFD,S
BE9D: 10 AF 61   STY   $0001,S
;
;-----;
;
; DoSoundTableIndexYStackSet()
;
BEA0: D6 11      LDB   $0011
BEA2: E7 E4      STB   ,S
BEA4: 8E 82 0A    LDX   #$820A
BEA7: BD AC 46   JSR   $AC46
BEAA: BD 90 76   JSR   $9076
BEAD: EC 04      LDD   $0004,X
BEAF: 85 04      BITA  #$04
BEB1: 27 07      BEQ   $BEB1
BEB3: CE 04 08   LDU   #$0408
```

BEB6: 8D 74	BSR \$BF2C
BEB8: 25 67	BCS \$BF21
BEBA: C5 08	BITB #\$08
BEBE: 27 07	BEQ \$BEC5
BEBE: CE 03 FC	LDU #\$03FC
BEC1: 8D 69	BSR \$BF2C
BEC3: 25 5C	BCS \$BF21
BEC5: C5 10	BITB #\$10
BEC7: 26 0A	BNE \$BED3
.	
.	
.	

You can compare the chunks of code above between The Shadow and Indiana Jones and see they do appear to be the same functions, so it is then likely to assume that here we have found the TS_LH6 SoundCallTable[] pointer, value 0x820A.

SoundCallTable[] Pointer

The previous sections revealed SoundCallTable[] pointers:

- 0x81F8 for IJ_L8
- 0x820A for TS_LH6

These are WPC addresses in the non-banked region of the ROM.

At these addresses we will find the actual WPC address of the SoundCallTable[].

- In IJ_L7 ROM, at WPC address 0x81F8 we find 3 bytes 0x43, 0x95, 0x30 representing WPC address \$4395,30
- In TS_LH6 ROM, at WPC address 0x820A we find 3 bytes 0x41, 0xE4, 0x35 representing WPC address \$41E4,35

SoundCallTable[]

Here we will show the SoundCallTable[] from IJ_L7. As we discovered, it is located at WPC address \$4395,30. Each table entry is 6 bytes in length. The meaning of all 6 bytes isn't fully understood at the time of this writing but here is what we have so far:

- Bytes [0][1], 16-bit number either the actual sound call number sent to sound card or address of sound commands (details later)
- Bytes[2][3], <not researched yet, tbd>
- Byte[4], Flag byte, specifies characteristics about how the sound is played (details tbd).
- Byte[5], Flag byte, specifies characteristics about how the sound is played (details tbd).
 - The 0x01 bit of Byte[5] indicates if Bytes[0][1] are an address of sound commands and not an actual sound number.

Below we will see the entire IJ_L7 SoundCallTable[]. Although earlier in this document we showed how to find The Shadow SoundCallTable[] it is not going to be shown here since it has a similar format as the IJ_L7 table:

```

;-----;-----
;
; SoundCallTable[]
;
; [0][1] = Sound Index to Play <or> Address of Sound call commands (when 0x01 bit of [5] byte is set)
;
; [2]
;
; [3]
;
; [4] 76543210
;      | | | | | | |
;      | | | | | | \-----
;      | | | | | | \-----
;      | | | | | | \----- When set, Sound played with U=0x0401, else (if bit [5] 0x80 not set) played with U=0x03E9
;      | | | | \-----
;      | | | \-----
;      | | \-----
;      | \-----
;      \-----
;
; [5] 76543210
;      | | | | | | |
;      | | | | | | \----- When set, this means [0][1] is address in ROM (this bank) of sound call commands
```

```

;      | | | | | \-----
;      | | | | \-----
;      | | | \-----
;      | | \-----
;      | \-----
;      | \-----
;      | \-----
;      \----- When set, Sound played with U=0x03F5, else (if bit[4] 0x04 not set) played with U=0x03E9
;
;
4395: 01 22      ; The number of table entries is 0x0122 (290 entries)
4397: 06          ; Each entry is 6-bytes in length
;
4398: 00 00      ; SoundTableEntry00, Sound00, Stop playing
439A: 00 10      ;
439C: 00 00      ;
;
439E: 00 71      ; SoundTableEntry01, Sound71, Swishy/sliding sound
43A0: 85 A5      ;
43A2: 00 28      ;
;
43A4: 00 4F      ; SoundTableEntry02, Sound4F, Screaming as mine-cart falls
43A6: B4 72      ;
43A8: 00 30      ;
;
43AA: 00 3F      ; SoundTableEntry03, Sound3F, Crash
43AC: 4A 72      ;
43AE: 00 28      ;
;
43B0: 03 DA      ; SoundTableEntry04, Sound3DA, WPC Menu tone
43B2: 01 10      ;
43B4: 04 20      ;
;
43B6: 03 D9      ; SoundTableEntry05, Sound3D9, WPC Menu tone
43B8: 01 10      ;
43BA: 04 20      ;
;
43BC: 00 64      ; SoundTableEntry06, Sound64, punch/disturbance
43BE: 40 50      ;
43C0: 00 30      ;
;
43C2: 00 65      ; SoundTableEntry07, Sound65, punch/disturbance
43C4: 40 50      ;
43C6: 00 30      ;
;
43C8: 00 7F      ; SoundTableEntry08, Sound7F, sliding sound

```

43CA: 4F 64	;
43CC: 00 28	;
	;
43CE: 02 1F	; SoundTableEntry09, Sound21F, disturbance/sliding
43D0: 54 65	;
43D2: 00 28	;
	;
43D4: 02 20	; SoundTableEntry0A, Sound220, gunshot
43D6: 72 68	;
43D8: 00 28	;
	;
43DA: 00 44	; SoundTableEntry0B, Sound44, gunshot with echo
43DC: 3D 71	;
43DE: 00 28	;
	;
43E0: 00 FF	; SoundTableEntry0C, SoundFF, gunshot with echo
43E2: 78 5B	;
43E4: 00 28	;
	;
43E6: 02 21	; SoundTableEntry0D, Sound221, arrow sound
43E8: 23 5D	;
43EA: 00 28	;
	;
43EC: 01 64	; SoundTableEntry0E, Sound164, effect, INDY rollover
43EE: 14 30	;
43F0: 04 20	;
	;
43F2: 01 65	; SoundTableEntry0F, Sound165, effect, INDY rollover
43F4: 16 4A	;
43F6: 04 20	;
	;
43F8: 00 4E	; SoundTableEntry10, Sound4E, gunshot hits
43FA: 45 45	;
43FC: 00 28	;
	;
43FE: 00 7B	; SoundTableEntry11, Sound7B, distant gunshot
4400: 3D 45	;
4402: 00 28	;
	;
4404: 02 22	; SoundTableEntry12, Sound222, gunshot
4406: 38 5D	;
4408: 00 28	;
	;
440A: 02 20	; SoundTableEntry13, Sound220, gunshot
440C: 72 5D	;

440E: 00 28	;
	;
4410: 00 41	; SoundTableEntry14, Sound41, airplane flying overhead
4412: 94 5A	;
4414: 04 20	;
	;
4416: 00 42	; SoundTableEntry15, Sound42, airplane flying overhead
4418: CC 5A	;
441A: 04 20	;
	;
441C: 00 EA	; SoundTableEntry16, SoundEA, airplane not starting
441E: 2A 5A	;
4420: 04 28	;
	;
4422: 00 EB	; SoundTableEntry17, SoundEB, airplane not starting
4424: 81 5A	;
4426: 04 28	;
	;
4428: 4A F4	; SoundTableEntry18, SoundCall14AF4, gunshot sounds
442A: 00 5A	;
442C: 04 29	;
	;
442E: 00 76	; SoundTableEntry19, Sound76, Punch
4430: 16 20	;
4432: 00 28	;
	;
4434: 00 77	; SoundTableEntry1A, Sound77, Punch
4436: 17 20	;
4438: 00 28	;
	;
443A: 00 78	; SoundTableEntry1B, Sound78, Punch
443C: 13 20	;
443E: 00 28	;
	;
4440: 00 79	; SoundTableEntry1C, Sound79, Punch
4442: 12 20	;
4444: 00 28	;
	;
4446: 00 60	; SoundTableEntry1D, Sound60, horse galloping by
4448: 89 5A	;
444A: 00 28	;
	;
444C: 00 61	; SoundTableEntry1E, Sound61, Vehicle driving by
444E: D1 5A	;
4450: 00 28	;

4452: 00 62	;
4454: A8 5A	; SoundTableEntry1F, Sound62, Motorcycle driving by
4456: 00 28	;
	;
4458: 00 63	; SoundTableEntry20, Sound63, Car driving by
445A: C5 5A	;
445C: 00 28	;
	;
445E: 00 50	; SoundTableEntry21, Sound50, Dragging, sliding sound
4460: 66 5A	;
4462: 00 28	;
	;
4464: 00 5B	; SoundTableEntry22, Sound5B, "I keep telling you, you listen me more, you live longer!"
4466: CD 5D	;
4468: 00 30	;
	;
446A: 00 58	; SoundTableEntry23, Sound58, "Indiana Jones, I knew some day you'd come walking back through
my door"	
446C: FF 5D	;
446E: 00 30	;
	;
4470: 00 59	; SoundTableEntry24, Sound59, "Holy smoke my friend, I'm so pleased you're not dead!"
4472: E6 5D	;
4474: 00 30	;
	;
4476: 03 E5	; SoundTableEntry25, Sound3E5, **nothing**
4478: FA 5D	;
447A: 00 30	;
	;
447C: 00 5C	; SoundTableEntry26, Sound5C, "Junior? It is you junior!"
447E: 87 5D	;
4480: 00 30	;
	;
4482: 00 7F	; SoundTableEntry27, Sound7F, Brief disturbance
4484: 4F 60	;
4486: 00 28	;
	;
4488: 00 67	; SoundTableEntry28, Sound67, Brief skid
448A: 2C 60	;
448C: 00 28	;
	;
448E: 00 3C	; SoundTableEntry29, Sound3C, Prop airplane flyby
4490: 76 5A	;
4492: 00 28	;

4494: 02 2D	;
4496: 7A 5A	; SoundTableEntry2A, Sound22D, Plane flyby
4498: 04 20	;
	;
449A: 4B 2C	; SoundTableEntry2B, SoundCall14B2C, Extra ball serve-up
449C: 00 6F	;
449E: 04 39	;
	;
44A0: 01 63	; SoundTableEntry2C, Sound163, effect, extra ball lit
44A2: 16 50	;
44A4: 04 20	;
	;
44A6: 00 6E	; SoundTableEntry2D, Sound6E, Burp
44A8: 3C 50	;
44AA: 00 28	;
	;
44AC: 4B 86	; SoundTableEntry2E, SoundCall14B86, hurryup achieved
44AE: 00 74	;
44B0: 04 31	;
	;
44B2: 4A D4	; SoundTableEntry2F, SoundCall14AD4, Jackpot achieved
44B4: 00 76	;
44B6: 04 39	;
	;
44B8: 4A DB	; SoundTableEntry30, SoundCall14ADB, Double-Jackpot achieved
44BA: 00 76	;
44BC: 04 39	;
	;
44BE: 4A E2	; SoundTableEntry31, SoundCall14AE2, Triple-Jackpot achieved
44C0: 00 76	;
44C2: 04 39	;
	;
44C4: 00 AD	; SoundTableEntry32, SoundAD, "Super jackpot!!!"
44C6: 0A 76	;
44C8: 04 38	;
	;
44CA: 02 24	; SoundTableEntry33, Sound224, gunshot
44CC: 54 5B	;
44CE: 00 28	;
	;
44D0: 4A 9E	; SoundTableEntry34, SoundCall14A9E, Mode Start sounds, Get the idol
44D2: 00 71	;
44D4: 04 31	;
	;

44D6: 00 8B	; SoundTableEntry35, Sound8B, Mode Start sound, Streets of Cairo, "Cairo, city of the living"
44D8: 6F 71	;
44DA: 00 30	;
	;
44DC: 00 8A	; SoundTableEntry36, Sound8A, Mode Start sound, Well of Souls, "Asps, very dangerous, you go
first"	
44DE: 73 71	;
44E0: 00 30	;
	;
44E2: 02 2C	; SoundTableEntry37, Sound22C, Mode Start sound, Raven Bar, "Kill them both"
44E4: 4E 71	;
44E6: 00 30	;
	;
44E8: 00 CE	; SoundTableEntry38, SoundCE, Mode Start sound, Monkey Brains, "Ahh, chilled monkey brains"
44EA: 8F 71	;
44EC: 00 30	;
	;
44EE: 4A A9	; SoundTableEntry39, SoundCall14AA9, Mode Start sounds, Steal the stones
44F0: 00 71	;
44F2: 00 31	;
	;
44F4: 00 E9	; SoundTableEntry3A, SoundE9, Mode Start sound, Rope Bridge, "Come on, Indy come on!"
44F6: 84 71	;
44F8: 00 30	;
	;
44FA: 4A B4	; SoundTableEntry3B, SoundCall14AB4, Mode Start sounds, Castle Greunwaldt
44FC: 00 71	;
44FE: 00 31	;
	;
4500: 4A C4	; SoundTableEntry3C, SoundCall14AC4, Mode Start sounds, Take chase
4502: 00 71	;
4504: 00 31	;
	;
4506: 00 9C	; SoundTableEntry3D, Sound9C, Mode Start sound, Three Challenges, "You're meddling with
powers you cannot possibly comprehend"	
4508: DC 71	;
450A: 00 30	;
	;
450C: 00 6B	; SoundTableEntry3E, Sound6B, Monkey scream
450E: 21 50	;
4510: 00 28	;
	;
4512: 00 6C	; SoundTableEntry3F, Sound6C, Monkey scream
4514: 18 50	;
4516: 00 28	;

4518: 00 6D	;
451A: 0C 50	; SoundTableEntry40, Sound6D, Monkey chirp
451C: 00 28	;
	;
451E: 00 6A	; SoundTableEntry41, Sound6A, Trump fanfare award
4520: D3 6B	;
4522: 04 20	;
	;
4524: 4A 64	; SoundTableEntry42, SoundCall14A64, Monkey brains, eating soup sounds
4526: 00 6B	;
4528: 00 29	;
	;
452A: 4A 79	; SoundTableEntry43, SoundCall14A79, Monkey brains, eating food sounds
452C: 00 6B	;
452E: 00 39	;
	;
4530: 4A 89	; SoundTableEntry44, SoundCall14A89, Monkey brains, drinking
4532: 00 6B	;
4534: 00 39	;
	;
4536: 00 7A	; SoundTableEntry45, Sound7A, Punch - ooh
4538: 0F 20	;
453A: 00 28	;
	;
453C: 00 52	; SoundTableEntry46, Sound52, Metal bang
453E: 26 6F	;
4540: 00 28	;
	;
4542: 00 64	; SoundTableEntry47, Sound64, Metal bang
4544: 40 6F	;
4546: 00 30	;
	;
4548: 00 44	; SoundTableEntry48, Sound44, Gun blast
454A: 3D 6F	;
454C: 00 28	;
	;
454E: 00 4C	; SoundTableEntry49, Sound4C, Punch
4550: 0D 20	;
4552: 00 28	;
	;
4554: 00 7D	; SoundTableEntry4A, Sound7D, Gunshot
4556: 19 45	;
4558: 00 28	;
	;

455A: 02 1D	; SoundTableEntry4B, Sound21D, Gunshots with echo
455C: 59 45	;
455E: 00 28	;
	;
4560: 01 2F	; SoundTableEntry4C, Sound12F, fanfare, mode-start
4562: B7 71	;
4564: 04 20	;
	;
4566: 00 4B	; SoundTableEntry4D, Sound4B, Woosh missile-like sound
4568: 35 71	;
456A: 00 28	;
	;
456C: 00 88	; SoundTableEntry4E, Sound88, Gun trigger click
456E: 15 20	;
4570: 00 28	;
	;
4572: 00 52	; SoundTableEntry4F, Sound52, Gunshot Ricochet
4574: 26 20	;
4576: 00 28	;
	;
4578: 00 53	; SoundTableEntry50, Sound53, Gunshot
457A: 28 20	;
457C: 00 28	;
	;
457E: 00 54	; SoundTableEntry51, Sound54, Gunshot
4580: 2D 20	;
4582: 00 28	;
	;
4584: 00 A5	; SoundTableEntry52, SoundAF, "Trippple Jackpot!"
4586: 1B 20	;
4588: 00 28	;
	;
458A: 00 8F	; SoundTableEntry53, Sound8F, Gunfire
458C: 41 45	;
458E: 00 28	;
	;
4590: 00 90	; SoundTableEntry54, Sound90, Gunfire
4592: 28 45	;
4594: 00 28	;
	;
4596: 00 A1	; SoundTableEntry55, SoundA1, Airplane flyby
4598: 9B 5A	;
459A: 00 30	;
	;
459C: 00 A4	; SoundTableEntry56, SoundA4, Airplane machine gun

459E: 46 5A	;
45A0: 04 28	;
	;
45A2: 4A FF	; SoundTableEntry57, SoundCall4AFF, coin-up sounds
45A4: 00 77	;
45A6: 00 29	;
	;
45A8: 01 2E	; SoundTableEntry58, Sound12E, sound, mine-cart falling
45AA: D1 A5	;
45AC: 04 20	;
	;
45AE: 00 49	; SoundTableEntry59, Sound49, Continuous: Mine-cart travel
45B0: 01 A5	;
45B2: 00 28	;
	;
45B4: 00 0C	; SoundTableEntry5A, Sound0C, Continuous: Mine-cart music
45B6: 3C A5	;
45B8: 04 20	;
	;
45BA: 00 4A	; SoundTableEntry5B, Sound4A, Mine-cart rails duing turn
45BC: 3D 72	;
45BE: 00 30	;
	;
45C0: 00 9D	; SoundTableEntry5C, Sound9D, "You must choose"
45C2: 4C 72	;
45C4: 00 30	;
	;
45C6: 00 9E	; SoundTableEntry5D, Sound9E, "Choose wisely"
45C8: 50 72	;
45CA: 00 30	;
	;
45CC: 00 9F	; SoundTableEntry5E, Sound9F, "You have chosen, poorly"
45CE: A5 72	;
45D0: 00 30	;
	;
45D2: 00 A0	; SoundTableEntry5F, SoundA0, "You have chosen, Wisely"
45D4: 4A 72	;
45D6: 00 30	;
	;
45D8: 00 84	; SoundTableEntry60, Sound84, Curdling man scream
45DA: 7B 72	;
45DC: 00 28	;
	;
45DE: 00 46	; SoundTableEntry61, Sound46, Walup swish
45E0: 09 72	;

45E2: 00 28	;
	;
45E4: 00 93	; SoundTableEntry62, Sound93, Soft punch?
45E6: 0B 6B	;
45E8: 00 28	;
	;
45EA: 00 94	; SoundTableEntry63, sound94, Soft punch?
45EC: 0B 6B	;
45EE: 00 28	;
	;
45F0: 00 95	; SoundTableEntry64, Sound95, Soft punch?
45F2: 0A 6B	;
45F4: 00 28	;
	;
45F6: 00 96	; SoundTableEntry65, Sound96, Soft punch?
45F8: 09 6B	;
45FA: 00 28	;
	;
45FC: 00 97	; SoundTableEntry66, Sound97, Soft punch?
45FE: 09 6B	;
4600: 00 28	;
	;
4602: 00 98	; SoundTableEntry67, Sound98, Soft punch?
4604: 09 6B	;
4606: 00 28	;
	;
4608: 00 99	; SoundTableEntry68, Sound99, Soft punch?
460A: 0A 6B	;
460C: 00 28	;
	;
460E: 02 27	; SoundTableEntry69, Sound227, Punch and heavy breathing
4610: 81 6B	;
4612: 00 30	;
	;
4614: 01 2C	; SoundTableEntry6A, Sound12C, peril effect
4616: DC 6B	;
4618: 04 20	;
	;
461A: 00 E0	; SoundTableEntry6B, SoundE0, Door slam
461C: 12 5A	;
461E: 04 28	;
	;
4620: 00 89	; SoundTableEntry6C, Sound89, "Carefully, carefully!"
4622: 59 80	;
4624: 00 30	;

4626: 4B 08	;
4628: 00 6B	; SoundTableEntry6D, SoundCall14B08, Streets of Chair, Marion found
462A: 00 31	;
	;
462C: 00 91	; SoundTableEntry6E, Sound91, "See ya tomorrow, Indiana Jones"
462E: 87 EF	;
4630: 00 38	;
	;
4632: 00 A3	; SoundTableEntry6F, SoundA3, Laugh
4634: 42 EF	;
4636: 00 30	;
	;
4638: 00 CB	; SoundTableEntry70, SoundCB, "No time to argue!"
463A: 32 6B	;
463C: 00 30	;
	;
463E: 00 CC	; SoundTableEntry71, SoundCC, "Throw me the idol, I throw you the whip!"
4640: 57 6B	;
4642: 04 20	;
	;
4644: 00 CA	; SoundTableEntry72, SoundCA, "Gimme the whip!"
4646: 1B 6B	;
4648: 00 30	;
	;
464A: 00 CD	; SoundTableEntry73, SoundCD, "Adios amigo"
464C: 39 6B	;
464E: 04 20	;
	;
4650: 00 85	; SoundTableEntry74, Sound85, "We could at least behave like civilized people"
4652: 73 F0	;
4654: 00 30	;
	;
4656: 00 B6	; SoundTableEntry75, SoundB6, "Get to the path of adventure, Indy"
4658: 6D 50	;
465A: 00 30	;
	;
465C: 00 EE	; SoundTableEntry76, SoundEE, Tilt horn
465E: 64 F0	;
4660: 00 28	;
	;
4662: 00 51	; SoundTableEntry77, Sound51, Avalanche
4664: FF 6B	;
4666: 00 28	;
	;

4668: 00 69	; SoundTableEntry78, Sound69, Thunderstrike
466A: 8F 30	;
466C: 04 28	;
	;
466E: 02 21	; SoundTableEntry79, Sound221, Arrow through air?
4670: 23 77	;
4672: 00 28	;
	;
4674: 00 C5	; SoundTableEntry7A, SoundC5, Bullwhip snap
4676: 24 77	;
4678: 00 28	;
	;
467A: 00 D0	; SoundTableEntry7B, SoundD0, Abrupt hiss
467C: 47 30	;
467E: 04 20	;
	;
4680: 00 57	; SoundTableEntry7C, Sound57, Rattlesnake
4682: 84 30	;
4684: 04 20	;
	;
4686: 00 3D	; SoundTableEntry7D, Sound3D, Punch swish
4688: 22 30	;
468A: 00 28	;
	;
468C: 00 3E	; SoundTableEntry7E, Sound3E, Woosh Effect
468E: 23 30	;
4690: 00 28	;
	;
4692: 00 CF	; SoundTableEntry7F, SoundCF, Gunshot
4694: 78 6B	;
4696: 00 28	;
	;
4698: 02 1C	; SoundTableEntry80, Sound21C, Horse Gallop
469A: 89 6B	;
469C: 00 28	;
	;
469E: 00 46	; SoundTableEntry81, Sound46, Bloop Sound
46A0: 09 30	;
46A2: 00 28	;
	;
46A4: 00 47	; SoundTableEntry82, Sound47, Stick through air swish High
46A6: 07 30	;
46A8: 00 28	;
	;
46AA: 00 48	; SoundTableEntry83, Sound48, Stick through air swish Low

46AC: 0A 30	;
46AE: 00 28	;
	;
46B0: 00 45	; SoundTableEntry84, Sound45, Bad guy laugh
46B2: 47 6B	;
46B4: 00 30	;
	;
46B6: 01 99	; SoundTableEntry85, Sound199
46B8: 32 6B	;
46BA: 04 20	;
	;
46BC: 01 9D	; SoundTableEntry86, Sound19D
46BE: 32 6B	;
46C0: 04 20	;
	;
46C2: 01 9A	; SoundTableEntry87, Sound19A
46C4: 32 6B	;
46C6: 04 20	;
	;
46C8: 01 9E	; SoundTableEntry88, Sound19E
46CA: 32 6B	;
46CC: 04 20	;
	;
46CE: 01 9B	; SoundTableEntry89, Sound19B
46D0: 32 6B	;
46D2: 04 20	;
	;
46D4: 01 9F	; SoundTableEntry8A, Sound19F
46D6: 32 6B	;
46D8: 04 20	;
	;
46DA: 01 9C	; SoundTableEntry8B, Sound19C
46DC: 32 6B	;
46DE: 04 20	;
	;
46E0: 01 A0	; SoundTableEntry8C, Sound1A0
46E2: 32 6B	;
46E4: 04 20	;
	;
46E6: 01 32	; SoundTableEntry8D, Sound132, Sallah, "Extra Ball!"
46E8: A9 6F	;
46EA: 00 30	;
	;
46EC: 01 31	; SoundTableEntry8E, Sound131, musical failure/Match negative
46EE: 46 6F	;

46F0: 04 20	;
	;
46F2: 00 86	; SoundTableEntry8F, Sound86, Scream as Indy falls into pit
46F4: 9A 6F	;
46F6: 00 30	;
	;
46F8: 00 AA	; SoundTableEntry90, SoundAA, Thunder
46FA: 36 30	;
46FC: 04 28	;
	;
46FE: 00 9B	; SoundTableEntry91, Sound9B, "You call this archaeology?"
4700: 4B 6B	;
4702: 00 30	;
	;
4704: 00 DB	; SoundTableEntry92, SoundDB, "I told you..."
4706: 43 6B	;
4708: 00 30	;
	;
470A: 00 4E	; SoundTableEntry93, Sound4E, Gunshots
470C: 45 6B	;
470E: 00 28	;
	;
4710: 00 DC	; SoundTableEntry94, SoundDC, "...don't call me junior"
4712: 63 6B	;
4714: 00 30	;
	;
4716: 00 64	; SoundTableEntry95, Sound64, knock/thud
4718: 0A 6B	;
471A: 00 30	;
	;
471C: 00 62	; SoundTableEntry96, Sound62, motorcycle driving by
471E: A8 6B	;
4720: 00 28	;
	;
4722: 00 D6	; SoundTableEntry97, SoundD6, "cmon dad"
4724: 3B 6B	;
4726: 00 30	;
	;
4728: 01 A1	; SoundTableEntry98, Sound1A1
472A: 1E EE	;
472C: 04 20	;
	;
472E: 01 A3	; SoundTableEntry99, Sound1A3
4730: 1E EE	;
4732: 04 20	;


```
4734: 01 A5
4736: 1E EE
4738: 04 20

473A: 01 A7
473C: 1E EE
473E: 04 20

4740: 01 A9
4742: 1E EE
4744: 04 20

4746: 01 AB
4748: 1E EE
474A: 04 20

474C: 01 AD
474E: 1E EE
4750: 04 20

4752: 01 AF
4754: 1E EE
4756: 04 20

4758: 01 B1
475A: 1E EE
475C: 04 20

475E: 01 B3
4760: 1E EE
4762: 04 20

4764: 01 B5
4766: 1E EE
4768: 04 20

476A: 01 B7
476C: 1E EE
476E: 04 20

4770: 01 BC
4772: 1E EE
4774: 04 20
```

```
;
; SoundTableEntry9A
;
;
;
; SoundTableEntry9B
;
;
;
; SoundTableEntry9C
;
;
;
; SoundTableEntry9D
;
;
;
; SoundTableEntry9E
;
;
;
; SoundTableEntry9F
;
;
;
; SoundTableEntryA0
;
;
;
; SoundTableEntryA1
;
;
;
; SoundTableEntryA2
;
;
;
; SoundTableEntryA3
;
;
;
; SoundTableEntryA4
;
;
```

4776: 01 33	; SoundTableEntryA5, Sound133, Trumpet fanfare
4778: 7C 6F	;
477A: 04 20	;
	;
477C: 01 34	; SoundTableEntryA6, Sound134, Trumpet fanfare (at bonus)
477E: 7A 6F	;
4780: 04 20	;
	;
4782: 01 30	; SoundTableEntryA7, Sound130, Trumpet fanfare (after mode)
4784: B4 70	;
4786: 04 20	;
	;
4788: 00 44	; SoundTableEntryA8, Sound44, Gun blast
478A: 3D 7A	;
478C: 00 28	;
	;
478E: 01 69	; SoundTableEntryA9, Sound169, Brief tone
4790: 18 7A	;
4792: 04 20	;
	;
4794: 01 6A	; SoundTableEntryAA, Sound16A, Brief tone
4796: 18 7A	;
4798: 04 20	;
	;
479A: 00 43	; SoundTableEntryAB, Sound43, Slam tilt buzz sound
479C: 3D F0	;
479E: 04 20	;
	;
47A0: 00 B0	; SoundTableEntryAC, SoundB0, "Hit the jackpot!"
47A2: 50 6A	;
47A4: 00 30	;
	;
47A6: 00 B1	; SoundTableEntryAD, SoundB1, "Get the super jackpot!"
47A8: 46 6A	;
47AA: 00 30	;
	;
47AC: 00 B9	; SoundTableEntryAE, SoundB9, Mine-cart steer left sound
47AE: 16 72	;
47B0: 04 20	;
	;
47B2: 00 BA	; SoundTableEntryAF, SoundBA, Mine-cart steer right sound
47B4: 16 72	;
47B6: 04 20	;
	;
47B8: 00 B4	; SoundTableEntryB0, SoundB4, "The prize is double!"

47BA: 41 6A	;
47BC: 00 30	;
	;
47BE: 00 B5	; SoundTableEntryB1, SoundB5, "The prize is triple!"
47C0: 53 6A	;
47C2: 00 30	;
	;
47C4: 00 D8	; SoundTableEntryB2, SoundD8, "The cross of Coronado"
47C6: 5F 60	;
47C8: 00 30	;
	;
47CA: 00 D9	; SoundTableEntryB3, SoundD9, "The diamond of Shanghai"
47CC: 60 60	;
47CE: 00 30	;
	;
47D0: 00 DA	; SoundTableEntryB4, SoundDA, "The fish of Tales"
47D2: 5F 60	;
47D4: 00 30	;
	;
47D6: 00 E2	; SoundTableEntryB5, SoundE2, "The idol of the Incas"
47D8: 7A 60	;
47DA: 00 30	;
	;
47DC: 00 E3	; SoundTableEntryB6, SoundE3, "The remains of Nirhaci"
47DE: 5B 60	;
47E0: 00 30	;
	;
47E2: 4A E9	; SoundTableEntryB7, SoundCall14AE9, Double-jackpot lit
47E4: 00 6A	;
47E6: 00 31	;
	;
47E8: 01 BB	; SoundTableEntryB8, Sound1BB, peril/anticipating effect
47EA: 78 4C	;
47EC: 00 28	;
	;
47EE: 01 BA	; SoundTableEntryB9, Sound1BA, Multiball start effect
47F0: FF 72	;
47F2: 04 20	;
	;
47F4: 01 75	; SoundTableEntryBA, Sound175, Fanfare, extra ball awarded
47F6: 78 40	;
47F8: 04 20	;
	;
47FA: 01 5E	; SoundTableEntryBB, Sound15E, Fanfare, INDY rollovers completed
47FC: 78 5D	;

47FE: 04 28	;
	;
4800: 01 36	; SoundTableEntryBC, Sound136, Effects, ball locked
4802: E6 60	;
4804: 04 20	;
	;
4806: 00 EF	; SoundTableEntryBD, SoundEF, gun ricochet
4808: 47 5A	;
480A: 04 20	;
	;
480C: 00 40	; SoundTableEntryBE, Sound40, elephant trumpet
480E: 50 77	;
4810: 00 28	;
	;
4812: 00 F5	; SoundTableEntryBF, SoundF5, "Don't touch anything"
4814: 99 6E	;
4816: 00 28	;
	;
4818: 01 37	; SoundTableEntryC0, Sound137, Fanfare, award
481A: 47 6F	;
481C: 04 20	;
	;
481E: 00 F0	; SoundTableEntryC1, SoundF0, large door knock
4820: 40 6E	;
4822: 00 30	;
	;
4824: 01 04	; SoundTableEntryC2, Sound104, "We have top men working on right now. who? Top men."
4826: F0 A0	;
4828: 00 38	;
	;
482A: 01 07	; SoundTableEntryC3, Sound107, Gunshot with echo
482C: E6 A0	;
482E: 04 20	;
	;
4830: 00 F7	; SoundTableEntryC4, SoundF7, dramatic horn sound
4832: 09 6F	;
4834: 00 28	;
	;
4836: 00 F8	; SoundTableEntryC5, SoundF8, dramatic horn sound
4838: 07 6F	;
483A: 00 28	;
	;
483C: 00 F9	; SoundTableEntryC6, SoundF9, dramatic horn sound
483E: 09 6F	;
4840: 00 28	;

4842: 00 7E	;
4844: 53 6E	; SoundTableEntryC7, Sound7E, Arrow hitting target
4846: 00 08	;
	;
4848: 00 DF	; SoundTableEntryC8, SoundDF, Arrows sound
484A: 20 6E	;
484C: 00 08	;
	;
484E: 00 F1	; SoundTableEntryC9, SoundF1, Arrow sound
4850: 2A 6E	;
4852: 00 08	;
	;
4854: 01 68	; SoundTableEntryCA, Sound168, Tone, POA rollover
4856: 18 20	;
4858: 04 20	;
	;
485A: 01 69	; SoundTableEntryCB, Sound169, Tone, POA rollover
485C: 16 20	;
485E: 04 20	;
	;
4860: 01 6A	; SoundTableEntryCC, Sound16A, Tone, POA rollover
4862: 14 20	;
4864: 04 20	;
	;
4866: 01 6B	; SoundTableEntryCD, Sound16B, Tone, POA rollover
4868: 14 20	;
486A: 04 20	;
	;
486C: 01 6C	; SoundTableEntryCE, Sound16C, Tone, POA rollover
486E: 16 20	;
4870: 04 20	;
	;
4872: 00 8D	; SoundTableEntryCF, Sound8D, Sallah "ahhh"
4874: 42 50	;
4876: 00 10	;
	;
4878: 4B 18	; SoundTableEntryD0, SoundCall14B18, Ball drained, "This is how we say goodbye in Germany"
487A: 00 50	;
487C: 00 19	;
	;
487E: 4B 23	; SoundTableEntryD1, SoundCall14B23, Dramatic failure effect
4880: 00 6B	;
4882: 04 29	;
	;

4884: 01 6D	; SoundTableEntryD2, Sound16D, Series of tones
4886: 46 50	;
4888: 04 20	;
	;
488A: 01 64	; SoundTableEntryD3, Sound164, Tones
488C: 14 4A	;
488E: 04 20	;
	;
4890: 00 F3	; SoundTableEntryD4, SoundF3, Musical transition effect
4892: 39 50	;
4894: 04 20	;
	;
4896: 01 66	; SoundTableEntryD5, Sound166, Opening music, Hand of Fate
4898: E2 75	;
489A: 04 20	;
	;
489C: 01 67	; SoundTableEntryD6, Sound167, Closing music, Hand of Fate
489E: 4D 75	;
48A0: 04 20	;
	;
48A2: 01 C0	; SoundTableEntryD7, Sound1C0, Match not awarded
48A4: 38 A0	;
48A6: 04 20	;
	;
48A8: 01 C2	; SoundTableEntryD8, Sound1C2, Match awarded
48AA: 3C A0	;
48AC: 04 20	;
	;
48AE: 00 C2	; SoundTableEntryD9, SoundC2, Whip sound, coin insert
48B0: 1E 77	;
48B2: 00 28	;
	;
48B4: 00 6B	; SoundTableEntryDA, Sound6B, Monkey scream
48B6: 1E 77	;
48B8: 00 28	;
	;
48BA: 00 A6	; SoundTableEntryDB, SoundA6, "Get the extra ball"
48BC: 41 50	;
48BE: 00 30	;
	;
48C0: 4B 35	; SoundTableEntryDC, SoundCall14B35, Three Challeneges achieved "The breath of God"
48C2: 00 6B	;
48C4: 00 19	;
	;
48C6: 4B 45	; SoundTableEntryDD, SoundCall14B45, Three Challeneges achieved "The word of God"

48C8: 00 6B	;
48CA: 00 19	;
	;
48CC: 4B 55	; SoundTableEntryDE, SoundCall14B55, Three Challeneges achieved "The path of God"
48CE: 00 6B	;
48D0: 00 19	;
	;
48D2: 01 09	; SoundTableEntryDF, Soundl09, Rattle sound, no credits
48D4: 16 77	;
48D6: 00 28	;
	;
48D8: 00 FF	; SoundTableEntryE0, SoundFF, gunshot with echo
48DA: 78 71	;
48DC: 00 28	;
	;
48DE: 00 F2	; SoundTableEntryE1, SoundF2, Gun sound with award
48E0: 78 6B	;
48E2: 04 38	;
	;
48E4: 01 00	; SoundTableEntryE2, Soundl00, Gun sound with echo
48E6: 78 6F	;
48E8: 00 28	;
	;
48EA: 4B 91	; SoundTableEntryE3, SoundCall14B91, Streets of Cairo basket open
48EC: 00 6B	;
48EE: 00 29	;
	;
48F0: 01 6E	; SoundTableEntryE4, Soundl6E, Series of tones
48F2: 1A 6B	;
48F4: 04 20	;
	;
48F6: 4B 65	; SoundTableEntryE5, SoundCall14B65, Minecart accomplished, Incredible
48F8: 00 6B	;
48FA: 04 21	;
	;
48FC: 4B 70	; SoundTableEntryE6, SoundCall14B70, Minecart accomplished, Well done
48FE: 00 6B	;
4900: 04 21	;
	;
4902: 4B 7B	; SoundTableEntryE7, SouncCall14B7B, Minecart accomplished, Excellent
4904: 00 6B	;
4906: 04 21	;
	;
4908: 01 71	; SoundTableEntryE8, Soundl71, Starry effect up
490A: 96 75	;

490C: 00 28	;
	;
490E: 01 72	; SoundTableEntryE9, Sound172, Starry effect down
4910: 78 75	;
4912: 00 28	;
	;
4914: 01 60	; SoundTableEntryEA, Sound160, "Well done my friend!" w/fanfare
4916: B4 70	;
4918: 04 20	;
	;
491A: 01 61	; SoundTableEntryEB, Sound161, "Excellent!" w/fanfare
491C: B4 70	;
491E: 04 20	;
	;
4920: 01 5F	; SoundTableEntryEC, Sound15F, "Incredible!" w/fanfare
4922: B4 70	;
4924: 04 20	;
	;
4926: 00 44	; SoundTableEntryED, Sound44, Gun blast
4928: 2D 6B	;
492A: 00 08	;
	;
492C: 01 65	; SoundTableEntryEE, Sound165, game tone
492E: 0E 6B	;
4930: 04 20	;
	;
4932: 01 78	; SoundTableEntryEF, Sound178, Narrow escape tones
4934: 0A 6B	;
4936: 04 20	;
	;
4938: 00 B7	; SoundTableEntryF0, SoundB7, "Well done my friend!"
493A: 5D 7A	;
493C: 00 30	;
	;
493E: 00 B8	; SoundTableEntryF1, SoundB8, "Excellent!"
4940: 4B 6B	;
4942: 00 30	;
	;
4944: 00 B7	; SoundTableEntryF2, SoundB7, "Well done my friend!"
4946: 5D 6B	;
4948: 00 30	;
	;
494A: 00 B3	; SoundTableEntryF3, SoundB3, "Incredible!"
494C: 44 6B	;
494E: 00 30	;

4950: 01 74	;
4952: 50 6B	; SoundTableEntryF4, Sound174, Series of tones, POA related
4954: 04 20	;
	;
4956: 01 91	; SoundTableEntryF5, Sound191, Series of tones, POA related
4958: 32 6B	;
495A: 04 20	;
	;
495C: 01 95	; SoundTableEntryF6, Sound195, Series of tones, POA related
495E: 32 6B	;
4960: 04 20	;
	;
4962: 01 92	; SoundTableEntryF7, Sound192, Series of tones, POA related
4964: 32 6B	;
4966: 04 20	;
	;
4968: 01 96	; SoundTableEntryF8, Sound196, Series of tones, POA related
496A: 32 6B	;
496C: 04 20	;
	;
496E: 01 93	; SoundTableEntryF9, Sound193, Series of tones, POA related
4970: 32 6B	;
4972: 04 20	;
	;
4974: 01 97	; SoundTableEntryFA, Sound197, Series of tones, POA related
4976: 32 6B	;
4978: 04 20	;
	;
497A: 01 94	; SoundTableEntryFB, Sound194, Series of tones, POA related
497C: 32 6B	;
497E: 04 20	;
	;
4980: 01 98	; SoundTableEntryFC, Sound198, Series of tones, POA related
4982: 32 6B	;
4984: 04 20	;
	;
4986: 01 03	; SoundTableEntryFD, Sound103, "You cheat, Dr. Jones!"
4988: 6E 6B	;
498A: 00 30	;
	;
498C: 01 05	; SoundTableEntryFE, Sound105, disturbance/sliding sound
498E: 5A 35	;
4990: 00 28	;
	;

4992: 01 06	; SoundTableEntryFF, Sound106, gunshot
4994: 50 35	;
4996: 00 28	;
	;
4998: 01 08	; SoundTableEntry100, Sound108, swoosh sound then screaming with echo
499A: F0 6B	;
499C: 04 20	;
	;
499E: 4B 9C	; SoundTableEntry101, SoundCall14B9C, Full whip sound and snap
49A0: 00 A0	;
49A2: 00 29	;
	;
49A4: 01 76	; SoundTableEntry102, Sound176, musical tones, POA related
49A6: 5A 6F	;
49A8: 04 20	;
	;
49AA: 4B A5	; SoundTableEntry103, SoundCall14BA5, "Jackpot!"
49AC: 00 76	;
49AE: 04 39	;
	;
49B0: 01 79	; SoundTableEntry104, Sound179, musical tones, POA related?
49B2: 3C 5B	;
49B4: 04 20	;
	;
49B6: 00 DD	; SoundTableEntry105, SoundDD, "Mole ram!"
49B8: 3E 52	;
49BA: 00 30	;
	;
49BC: 00 DE	; SoundTableEntry106, SounddE, "Prepare to meet Kali"
49BE: 73 52	;
49C0: 00 30	;
	;
49C2: 00 56	; SoundTableEntry107, Sound56, "You dare not do that"
49C4: AE 52	;
49C6: 00 30	;
	;
49C8: 02 2F	; SoundTableEntry108, Sound022F, "Snakes, why did it have to be snakes?"
49CA: 90 52	;
49CC: 00 30	;
	;
49CE: 00 F6	; SoundTableEntry109, SoundF6, "Should have mailed it to the Marx brothers. Will you take it
easy?"	
49D0: CB 52	;
49D2: 00 30	;
	;

49D4: 00 A8	; SoundTableEntry10A, SoundA8, subtle laugh
49D6: 50 52	;
49D8: 00 30	;
	;
49DA: 02 2B	; SoundTableEntry10B, Sound22B, eating, slurping
49DC: 58 52	;
49DE: 00 30	;
	;
49E0: 02 2E	; SoundTableEntry10C, Sound22E, "My medalion!"
49E2: 32 A0	;
49E4: 00 10	;
	;
49E6: 01 0C	; SoundTableEntry10D, Sound10C, Banging sound
49E8: 0A A0	;
49EA: 00 28	;
	;
49EC: 01 0B	; SoundTableEntry10E, Sound10B, Punch
49EE: 12 A0	;
49F0: 00 30	;
	;
49F2: 01 38	; SoundTableEntry10F, Sound138, Musical tones, award
49F4: 3C A0	;
49F6: 04 00	;
	;
49F8: 4B AC	; SoundTableEntry110, SoundCall14BAC, "Extra ball!" awarded
49FA: 00 A0	;
49FC: 04 11	;
	;
49FE: 4B B7	; SoundTableEntry111, SoundCall14BB7, "Eternal life, Dr. Jones"
4A00: 00 71	;
4A02: 00 19	;
	;
4A04: 01 00	; SoundTableEntry112, Sound100, Gun sound with echo
4A06: 78 70	;
4A08: 00 28	;
	;
4A0A: 01 3A	; SoundTableEntry113, Sound13A, "BILLION!!!!"
4A0C: F0 76	;
4A0E: 04 18	;
	;
4A10: 01 39	; SoundTableEntry114, Sound139, "Collect everything!"
4A12: 78 71	;
4A14: 00 10	;
	;
4A16: 00 44	; SoundTableEntry115, Sound44, Gun blast

4A18: 2D 5B	;
4A1A: 00 08	;
	;
4A1C: 00 76	; SoundTableEntry116, Sound76, Punch
4A1E: 0A 6B	;
4A20: 00 08	;
	;
4A22: 00 4B	; SoundTableEntry117, Sound4B, swoosh sound, during extra ball buy-in
4A24: 0A 5B	;
4A26: 00 08	;
	;
4A28: 01 0E	; SoundTableEntry118, Sound10E, Frog ribbit
4A2A: 0A 5B	;
4A2C: 00 08	;
	;
4A2E: 00 E6	; SoundTableEntry119, SoundE6, "Don't touch anything"
4A30: 50 5B	;
4A32: 00 10	;
	;
4A34: 00 AC	; SoundTableEntry11A, SoundAC, "Quickly"
4A36: 3C 5A	;
4A38: 00 10	;
	;
4A3A: 00 3C	; SoundTableEntry11B, Sound3C, Prop airplane flyby
4A3C: 76 1F	;
4A3E: 00 28	;
	;
4A40: 02 2D	; SoundTableEntry11C, Sound22D, Plane flyby overhead
4A42: 7A 1F	;
4A44: 04 20	;
	;
4A46: 00 41	; SoundTableEntry11D, Sound41, airplane flying overhead
4A48: 94 1F	;
4A4A: 04 20	;
	;
4A4C: 00 42	; SoundTableEntry11E, Sound42, airplane flying overhead
4A4E: CC 1F	;
4A50: 04 20	;
	;
4A52: 00 A1	; SoundTableEntry11F, SoundA1, Airplane flyby
4A54: 9B 1F	;
4A56: 00 30	;
	;
4A58: 00 A4	; SoundTableEntry120, SoundA4, Airplane machine gun
4A5A: 46 1F	;

```

4A5C: 04 28      ;
                ;
4A5E: 00 E0      ; SoundTableEntry121, SoundE0, Door slam
4A60: 12 1F      ;
4A62: 04 28      ;
                ;
;-----;-----
                ;

```

You can see most of the SoundCallTable[] entries contain a 16-bit sound number as the first 2 bytes of each entry. You should be able to see how you can edit the sound numbers to change or inhibit sounds in your ROM by modifying the particular sound's 6-byte table entry. For example you could replace an offensive sound number with one that's more generic or a sound number that results in no sounds at all (ie a "null" sound call).

You can also see that some of the SoundCallTable[] entries contain a WPC address instead of the sound number in the first two bytes of the SoundCallTable[] entry. These WPC addresses, I've noticed in U_L7 and in TS_LH6 (and presumably all other WPC games) point to bytes immediately after the SoundCallTable[] itself. These sound call addresses point to a series of sound call commands which we will show next:

Sound Call Commands

As mentioned above, some SoundCallTable[] entries, instead of having a 16-bit sound number, have a 16-bit WPC address which points to a series of Sound Call Commands.

Below we will see the entire set of Sound Call Commands from IJ_L7. They contain a series of commands which will run until the <end> marker is hit:

- 0x01 <0xyy>, Delay for 8-bit time value yy.
- 0x02 <0xyyyy>, Delay for 16-bit time value yyyy.
- 0x03 <end>
- 0x04 <0xyyyy>, Play sound number yyyy
- 0x05 <0xyyyy>, Play sound number yyyy
- 0x0D <0xyyyy>, Play sound number yyyy

Note the commands 0x04, 0x05 and 0x0D play a 16-bit sound number using different characteristics which I've yet to study and fully understand. Also note that when I went through TS_LH6 it only used command 0x04, so it's likely other games use different sorts of commands here but this should be a good starting point for your investigation.

```

;
;-----;
;
;
; Basically, the sound call table, above, can usually play a single, specified sound
; or it can point to memory below which is a series of sound calls.
;
;
;-----
; SoundTableEntry42, SoundCall14A64, Monkey brains, eating soup sounds
;-----
4A64: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A65: 00 70    ;   Sound70, slurping soup
4A67: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A68: 1A      ;   TimerCounter: 0x1A
4A69: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A6A: 00 73    ;   Sound73, Eating/slurping
4A6C: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A6D: 50      ;   TimerCounter: 0x50
```

```

4A6E: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4A6F: 00 55       ;   Sound55, swallowing
4A71: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A72: 14          ;   TimerCounter: 0x14
4A73: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A74: 00 6E       ;   Sound6E, burp
4A76: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A77: 3C          ;   TimerCounter: 0x3C
4A78: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntry43, SoundCall4A79, Monkey brains, eating food sounds
;-----;
4A79: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A7A: 00 73       ;   Sound73, Eating/slurping
4A7C: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A7D: 50          ;   TimerCounter: 0x50
4A7E: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4A7F: 00 55       ;   Sound55, swallowing
4A81: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A82: 14          ;   TimerCounter: 0x14
4A83: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A84: 00 6E       ;   Sound6E, burp
4A86: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A87: 5A          ;   TimerCounter: 0x5A
4A88: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntry44, SoundCall4A89, Monkey brains, drinking
;-----;
4A89: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A8A: 00 70       ;   Sound70, sipping drink
4A8C: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A8D: 32          ;   TimerCounter: 0x32
4A8E: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A8F: 00 73       ;   Sound73, Eating/slurping
4A91: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A92: 50          ;   TimerCounter: 0x50
4A93: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4A94: 00 55       ;   Sound55, swallowing
4A96: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.

```

```

4A97: 14          ;   TimerCounter: 0x14
4A98: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4A99: 00 6E       ;   Sound6E, burp
4A9B: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4A9C: 5A          ;   TimerCounter: 0x5A
4A9D: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntry34, SoundCall14A9E, Mode Start sounds, Get the idol
;-----;
4A9E: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4A9F: 00 87       ;   Sound87, "There's nothing to fear here"
4AA1: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AA2: 78          ;   TimerCounter: 0x78
4AA3: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4AA4: 00 8C       ;   Sound8C, "That's what scares me"
4AA6: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AA7: 4D          ;   TimerCounter: 0x4D
4AA8: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntry39, SoundCall14AA9, Mode Start sounds, Steal the stones
;-----;
4AA9: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4AAA: 02 2A       ;   Sound22A, "The stones are mine!"
4AAC: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AAD: 78          ;   TimerCounter: 0x78
4AAE: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4AAF: 00 A2       ;   SoundA2, "What a vivid imagination"
4AB1: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AB2: 78          ;   TimerCounter: 0x78
4AB3: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntry3B, SoundCall14AB4, Mode Start sounds, Castle Grunewald
;-----;
4AB4: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4AB5: 00 D2       ;   SoundD2, "I came here to save you!"
4AB7: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.

```



```

4AB8: 82          ;   TimerCounter: 0x82
4AB9: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4ABA: 00 D4       ;   SoundD4, "and who's going to come to save you?"
4ABC: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4ABD: 64          ;   TimerCounter: 0x64
4ABE: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4ABF: 00 D5       ;   SoundD5, "Junior!"
4AC1: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AC2: 3C          ;   TimerCounter: 0x3C
4AC3: 03          ; End of Commands
;
;-----
;
;-----
; SoundTableEntry3C, SoundCall14AC4, Mode Start sounds, Take chase
;-----
4AC4: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4AC5: 00 82       ;   Sound82, "Where's my father?"
4AC7: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AC8: 46          ;   TimerCounter: 0x46
4AC9: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4ACA: 00 8E       ;   Sound8E, "They have him..."
4ACC: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4ACD: 28          ;   TimerCounter: 0x28
4ACE: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4ACF: 00 BC       ;   SoundBC, "...in the belly of that steel beast"
4AD1: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AD2: 78          ;   TimerCounter: 0x78
4AD3: 03          ; End of Commands
;
;-----
;
;-----
; SoundTableEntry2F, SoundCall14AD4, Jackpot achieved
;-----
4AD4: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4AD5: 00 AB       ;   SoundAB, Sallah, "Jackpot" with music
4AD7: 02          ; Indexed function 0x02 Wait for the following 16-bit number of timer interrupt ticks.
4AD8: 01 54       ;   TimerCounter: 0x0154
4ADA: 03          ; End of Commands
;
;-----
;
;-----
; SoundTableEntry30, SoundCall14ADB, Double-Jackpot achieved

```

```

;-----
4ADB: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4ADC: 00 AE      ;   SoundAE, Sallah, "Double Jackpot" with music
4ADE: 02      ; Indexed function 0x02 Wait for the following 16-bit number of timer interrupt ticks.
4ADF: 01 68      ;   TimerCounter: 0x0168
4AE1: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntry31, SoundCall14AE2, Triple-Jackpot achieved
;-----
4AE2: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4AE3: 00 AF      ;   SoundAF, Sallah, "Triple Jackpot" with music
4AE5: 02      ; Indexed function 0x02 Wait for the following 16-bit number of timer interrupt ticks.
4AE6: 01 72      ;   TimerCounter: 0x0172
4AE8: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntryB7, SoundCall14AE9, Double-jackpot lit
;-----
4AE9: 05      ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4AEA: 00 B0      ;   SoundB0, "Hit the jackpot!"
4AEC: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AED: 64      ;   TimerCounter: 0x64
4AEE: 05      ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4AEF: 00 B4      ;   SoundB4, "The prize is double"
4AF1: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AF2: 3C      ;   TimerCounter: 0x3C
4AF3: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntry18, SoundCall14AF4, gunshot sounds
;-----
4AF4: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4AF5: 00 4D      ;   Sound4D, Gunshot
4AF7: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4AF8: 14      ;   TimerCounter: 0x14
4AF9: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4AFA: 00 E1      ;   SoundE1, Gunshot and dirt sliding sound
4AFC: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.

```

```

4AFD: 78          ;   TimerCounter: 0x78
4AFE: 03          ; End of Commands
;
;-----
;
;-----
; SoundTableEntry57, SoundCall14AFF, coin-up sounds
;-----
4AFF: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B00: 00 C2       ;   SoundC2, Whip sound, coin insert
4B02: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B03: 1E          ;   TimerCounter: 0x1E
4B04: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B05: 00 C5       ;   SoundC5, Bullwhip snap
4B07: 03          ; End of Commands
;
;-----
;
;-----
; SoundTableEntry6D, SoundCall14B08, Streets of Chair, Marion found
;-----
4B08: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B09: 00 92       ;   Sound92, "least you haven't forgotten how to show a lady a good time!"
4B0B: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B0C: C8          ;   TimerCounter: 0x01
4B0D: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B0E: 00 45       ;   Sound45, subtle laugh
4B10: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B11: 82          ;   TimerCounter: 0x82
4B12: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B13: 02 28       ;   Sound228, "oh shi#"
4B15: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B16: 5A          ;   TimerCounter: 0x5A
4B17: 03          ; End of Commands
;
;-----
;
;-----
; SoundTableEntryD0, SoundCall14B18, Ball drained, "This is how we say goodbye in Germany"
;-----
4B18: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B19: 00 D1       ;   SoundD1, "This is how we say goodbye in Germany"
4B1B: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B1C: 96          ;   TimerCounter: 0x96
4B1D: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008

```

```

4B1E: 00 76      ; Sound76, punch
4B20: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B21: 1E          ; TimerCounter: 0x1E
4B22: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntryD1, SoundCall14B23, Dramatic failure effect
;-----;
4B23: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B24: 01 77      ; Sound177, Dramatic music effect
4B26: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B27: 01 6F      ; Sound16F, Failure music (mine cart falling)
4B29: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B2A: 0A          ; TimerCounter: 0x0A
4B2B: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntry2B, SoundCall14B2C, Extra ball serve-up
;-----;
4B2C: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B2D: 00 BB      ; SoundBB, crashing sound
4B2F: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B30: 28          ; TimerCounter: 0x28
4B31: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B32: 01 32      ; Sound132, Sallah, "Extra Ball!"
4B34: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntryDC, SoundCall14B35, Three Challeneges achieved "The breath of God"
;-----;
4B35: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B36: 00 44      ; Sound44, gunshot
4B38: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B39: 1E          ; TimerCounter: 0x1E
4B3A: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B3B: 01 73      ; Sound173, Award fanfare
4B3D: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B3E: 5A          ; TimerCounter: 0x5A
4B3F: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010

```

```

4B40: 00 FC          ; SoundFC, "The breath of God"
4B42: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B43: 3C          ; TimerCounter: 0x3C
4B44: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntryDD, SoundCall14B45, Three Challeneges achieved "The word of God"
;-----;
4B45: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B46: 00 44        ; Sound44, gunshot
4B48: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B49: 1E          ; TimerCounter: 0x1E
4B4A: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B4B: 01 73        ; Sound173, Award fanfare
4B4D: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B4E: 5A          ; TimerCounter: 0x5A
4B4F: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B50: 00 FD        ; SoundFD, "The word of God"
4B52: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B53: 3C          ; TimerCounter: 0x3C
4B54: 03          ; End of Commands
;
;-----;
;
;-----;
; SoundTableEntryDE, SoundCall14B55, Three Challeneges achieved "The path of God"
;-----;
4B55: 04          ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B56: 01 00        ; Sound100, Gun blast
4B58: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B59: 32          ; TimerCounter: 0x32
4B5A: 0D          ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B5B: 01 73        ; Sound173, Award fanfare
4B5D: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B5E: 5A          ; TimerCounter: 0x5A
4B5F: 05          ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B60: 00 FE        ; SoundFE, "The path of God"
4B62: 01          ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B63: 3C          ; TimerCounter: 0x3C
4B64: 03          ; End of Commands
;
;-----;
;

```

```

;-----
; SoundTableEntryE5, SoundCall14B65, Minecart accomplished, Incredible
;-----
4B65: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B66: 01 70    ;   Sound170, Minecart breakthrough
4B68: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B69: C8      ;   TimerCounter: 0xC8
4B6A: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B6B: 01 5F    ;   Sound15F, "Incredible" with fanfare
4B6D: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B6E: B4      ;   TimerCounter: 0xB4
4B6F: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntryE6, SoundCall14B70, Minecart accomplished, Well done
;-----
4B70: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B71: 01 70    ;   Sound170, Minecart breakthrough
4B73: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B74: C8      ;   TimerCounter: 0xC8
4B75: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B76: 01 60    ;   Sound160, "Well done, my friend!"
4B78: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B79: B4      ;   TimerCounter: 0xB4
4B7A: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntryE7, SoundCall14B7B, Minecart accomplished, Excellent
;-----
4B7B: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B7C: 01 70    ;   Sound170, Minecart breakthrough
4B7E: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B7F: C8      ;   TimerCounter: 0xC8
4B80: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B81: 01 61    ;   Sound161, "Excellent!" with fanfare
4B83: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B84: B4      ;   TimerCounter: 0xB4
4B85: 03      ; End of Commands
;
;-----
;

```

```

;-----
; SoundTableEntry2E, SoundCall14B86, hurryup achieved
;-----
4B86: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4B87: 02 25    ;   Sound225, Airplane crashing into mountain
4B89: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B8A: B4      ;   TimerCounter: 0xB4
4B8B: 05      ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4B8C: 00 B2    ;   SoundB2, Sallah, "Great Shot!"
4B8E: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B8F: 3C      ;   TimerCounter: 0x3C
4B90: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntryE3, SoundCall14B91, Streets of Cairo basket open
;-----
4B91: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B92: 01 01    ;   Sound101, Streets of Cairo, basket up
4B94: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B95: 32      ;   TimerCounter: 0x32
4B96: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B97: 01 02    ;   Sound102, Streets of Cairo, basket down
4B99: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4B9A: 3C      ;   TimerCounter: 0x3C
4B9B: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntry101, SoundCall14B9C, Full whip sound and snap
;-----
4B9C: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4B9D: 02 21    ;   Sound221, Whip sound through air
4B9E: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4BA0: 1E      ;   TimerCounter: 0x1E
4BA1: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4BA2: 02 22    ;   Sound222, Whip snap
4BA3: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntry103, SoundCall14BA5, "Jackpot!"

```

```

;-----
4BA5: 0D      ; Indexed function 0x0D Put the next 2 bytes into circular sound buffer,D=0x0400
4BA6: 00 AB    ;   SoundAB, "Jackpot" with explosions
4BA8: 02      ; Indexed function 0x02 Wait for the following 16-bit number of timer interrupt ticks.
4BA9: 01 54    ;   TimerCounter: 0x154
4BAB: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntry110, SoundCall4BAC, "Extra ball!" awarded
;-----
4BAC: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4BAD: 00 44    ;   Sound44, Gun blast
4BAF: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4BB0: 28      ;   TimerCounter: 0x28
4BB1: 05      ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4BB2: 01 32    ;   Sound132, "Extra Ball" awarded
4BB4: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4BB5: A9      ;   TimerCounter: 0xA9
4BB6: 03      ; End of Commands
;
;-----
;
;-----
; SoundTableEntry111, SoundCall4BB7, "Eternal life, Dr. Jones"
;-----
4BB7: 04      ; Indexed function 0x04 Put the next 2 bytes into circular sound buffer,D=0x0008
4BB8: 01 00    ;   Sound100, Explosion
4BBA: 01      ; Indexed function 0x01 Wait for the following 8-bit number of timer interrupt ticks.
4BBB: 6E      ;   TimerCounter: 0x6E
4BBC: 05      ; Indexed function 0x05 Put the next 2 bytes into circular sound buffer,D=0x0010
4BBD: 00 72    ;   Sound72, "Eternal life, Dr. Jones"
4BBF: 03      ; End of Commands
;
;-----
;

```

You can see that in addition to the sound number in the SoundCallTable[] there is also the Sound Commands section, above, which may also contain a particular sound number in which you may be interested.

DoSoundTableParameterByte()

Now let's see how to find where your game might actually try to play these sounds. First thing you should keep in mind is that each SoundCallTable[] entry has a particular index. This is just the 0-based index in the table and not the actual sound number. Here we will see functions the WPC code will use to play a desired SoundCallTable[] by specifying its table index.

```
;-----;-----  
;  
; DoSoundTableParameterByte()  
;  
8599: 34 12      PSHS  X,A          ;  
859B: AE 63      LDX   $0003,S      ;  
859D: A6 80      LDA   ,X+          ; A gets 1-byte parameter  
859F: AF 63      STX   $0003,S      ;  
85A1: BD BF 1D    JSR   $BF1D        ; DoSoundTableIndexA()  
85A4: 35 92      PULS  A,X,PC       ; Done, RTS  
;  
;-----;
```

The function DoSoundTableParameterByte() is used for WPC code to play any SoundCallTable[] index from 0x00 through 0xFF. If you can find this function in your non-banked ROM then you can use its address to find places in the code that plays a desired sound number.

For example, in UJ_L7, we see in the SoundCallTable[], above, the sound index for burp sound effect is 0x6E. Now to find places in the code that plays the burp we will search the ROM for BD 85 99 6E, which is the JSR \$8599 with 0x6E parameter byte.

DoSoundTableParameterByte16Bit()

This is the 16-bit version which allows the caller to provide 2-bytes after the JSR opcode. This is used by WPC code to play a SoundCallTable[] index greater than 0xFF, thus needing 2 bytes to contain the SoundCallTable[] index (although it could also play sound call indexes <= 0xFF but it would be wasting a byte of ROM to do so).

```
;-----;-----  
;  
; DoSoundTableParameterByte16Bit()  
;  
88CA: 34 30      PSHS  Y,X          ;  
88CC: AE 64      LDX   $0004,S      ; X gets 2 bytes after function call  
88CE: 10 AE 81    LDY   ,X++        ; Y gets whatever X pointed to, x++  
88D1: AF 64      STX   $0004,S      ; Fixup return stack  
88D3: BD BF 2A    JSR   $BF2A        ; DoSoundTableIndexY()  
88D6: 35 B0      PULS  X,Y,PC       ; Done, RTS  
;  
;-----;
```

For example, in U_L7 the SoundCallTable[] index for “Quickly!” is 0x11A. This means we would search the ROM for BD 88 CA 01 1A to find the code that plays the “Quickly!” sound call. You should be able to find this function in your game by searching for byte patterns and also if you’ve found the address of “DoSoundTableIndexY()” then you can also search for the JSR to that function in your non-banked ROM in order to find this function.

DoSoundTableIndexD()

I've also noticed this function in IJ_L7. It will play the SoundTableIndex[] stored in the D register. The caller must have loaded the D register with the desired SoundCallTable[] index prior to calling this function. I'm not sure exactly under what circumstances this function would be used in lieu of the others but if you are having trouble finding when/how a particular sound number is being invoked, it may be via this function:

```
;-----;-----  
;  
; DoSoundTableIndexD()  
;  
B954: 34 20      PSHS  Y  
B956: 10 83 00 00 CMPD  #$0000  
B95A: 27 05      BEQ   $B961  
B95C: 1F 02      TFR   D,Y  
B95E: BD BF 2A   JSR   $BF2A      ; DoSoundTableIndexY()  
B961: 35 A0      PULS  Y,PC  
;  
;-----;
```

It's possible this function is handy for WPC code to use when loading a sound number from a table of sound numbers into D and then calling this function.

Playing Sound Based on Family Mode

Please refer to the other write-up “WPCAdjustments.pdf” where it describes the function “LookupGameAdjustmentParameter1andCheckIfEqualsParameter2 ()”. This is the function that WPC Code would use to determine if Family Mode is enabled. In IJ_L7 this function is at WPC address \$86AE. Since there is no “Family Mode” in IJ_L7, let’s shift to TS_LH6. By searching for the function in TS_LH6 non-banked ROM, I’ve found the address of LookupGameAdjustmentParameter1andCheckIfEqualsParameter2 () is at \$8709.

Also, using “WPCAdjustments.pdf” you can trace out the adjustments table to determine the index for “Family Mode” or you can simply load up the game and go into the adjustment menu and go to “Family Mode” and observe the DMD shows the index number, in the case of TS_LH6 it is at adjustment 28 (hex 0x1C).

This means, to find code that checks for family mode in TS_LH6 we would search for BD 87 09 1C. Searching the ROM for this pattern we can see how WPC code checks if Family Mode is enabled and plays a sound depending on the Family Mode setting.

Looking at the TS_LH6 ROM at offset 0x56AEB we have:

- BD 87 09 1C 01, Calls LookupGameAdjustmentParameter1andCheckIfEqualsParameter2 () to see if “Family Mode” is on.
- 24 06, C-bit clear, skips over 6 bytes. C-bit clear when adjustment is equal to 0x01 (C-bit clear when Family Mode is on).
- BD 85 F4 AA, Plays sound number 0xAA which I have traced to be “Let’s get the hell out of here!” Plays when Family Mode is off.
- 20 04, BRA, always skip over the following 4 bytes.
- BD 85 F4 E5, Plays sound number 0xE5 which I have traced to be “Ugh!” grunt sound. Plays when Family Mode is on.

Note that although the above has shown TS_LH6 checks Family Mode when deciding whether to play certain sounds, I’ve been told it still plays offensive sounds during Punish the Guilty mode even when Family Mode is On, so I believe that the Punish the Guilty mode uses a method other than direct BD 85 F4 to play sounds. It likely has a lookup table of SoundCallTable[] indexes and then calls DoSoundTableIndexD(), but this is just a guess.

PlayRandomSoundEffect() (IJ_L7)

This function gets called during attract mode on IJ_L7 when flipper buttons get hit. This function ends up playing a random sound call at certain times during attract mode. I'm showing this function here so you can possibly find similar function in your game and then you can modify the table of SoundCallTable[] indexes, which this function uses to select a random sound number.

I've found a similar function in TS_LH6 and will show that later in this document, so this means this code isn't specific to IJ_L7 and if you search for the right byte patterns you might be able to find a similar looking function in your game:

```
;-----;
;
; PlayRandomSoundEffect ()
;
; Called whenever left flipper button is pressed from $4A22,36
; Called whenever right flipper button is pressed from $49DC,36
;
4A5D: 34 76      PSHS  U,Y,X,B,A      ;
4A5F: 0D 8D      TST   $8D            ; if ($8D == 0) // $8D == 0 during game mode
4A61: 27 6E      BEQ   $4AD1          ; {
;     return;
; }
;
4A63: BD 86 AE    JSR   $86AE          ; LookupGameAdjustmentParameterlandCheckIfEqualsParameter2() // C-bit set when not-equal
4A66: 9A          ; 0x9A, Adjustments, Standard Adjustments, Tournament Play
4A67: 01          ; 0x01, "YES"
;
4A68: 25 0B      BCS   $4A75          ; if (TournamentPlay == "YES")
4A6A: 86 37      LDA   #$37           ; {
4A6C: B7 0A AD    STA   $0AAD          ; $0AAD = 0x37
;
4A6F: BD 85 A6    JSR   $85A6          ; RunFunctionIndexByteParameterFromTable74052F()
4A72: 04          ; TableEntry04, AttractModePerform() at $6D4D,36
4A73: 20 5C      BRA   $4AD1          ; return;
; }
;
4A75: BD 86 E3    JSR   $86E3          ;
4A78: 00 F0          ; Parameter 0x00F0
4A7A: 24 55      BCC   $4AD1          ;
4A7C: BD 9B 99    JSR   $9B99          ; Fetch59PtrLinkedListOffset11Word()
;
4A7F: 10 8C 00 85 CMPY  #$0085        ; if (Y == 0x0085)
4A83: 27 4C      BEQ   $4AD1          ; {
```

				; return;
				; }
				;
4A85: BD 86 AE	JSR	\$86AE		; LookupGameAdjustmentParameterlandCheckIfEqualsParameter2() // C-bit set when not-equal
4A88: 1E				; 0x1E, Adjustments, Feature Adjustments, Attract Mode Music
4A89: 01				; 0x01, "ON"
				;
4A8A: 25 45	BCS	\$4AD1		; if (AttractModeMusic != "ON")
				; {
				; return;
				; }
				;
4A8C: B6 0A B8	LDA	\$0AB8		; if (\$0AB8 != 0)
4A8F: 26 40	BNE	\$4AD1		; {
				; return;
				; }
				;
4A91: BD CB A7	JSR	\$CBA7		; AGets043A()
4A94: 81 F3	CMPA	#F3		; if (\$043A == 0xF3)
4A96: 27 39	BEQ	\$4AD1		; {
				; return;
				; }
				;
4A98: 86 7B	LDA	#\$7B		; A = 0x7B
4A9A: BD CB CF	JSR	\$CBCF		; Aminus043Cminus1()
4A9D: 24 32	BCC	\$4AD1		; if (A >= 0)
				; {
				; return;
				; }
				;
				;
4A9F: B6 1D B3	LDA	\$1DB3		; A = \$1DB3
4AA2: BD 89 48	JSR	\$8948		; CallBankedFunction_Param_WPCAddr()
4AA5: 68 6D 20				; ?
4AA8: 8E 4A DE	LDX	#\$4ADE		; X=0x4ADE
				;
4AAB: 86 0E	LDA	#\$0E		;--\ A gets 0x0E
4AAD: BD A7 D7	JSR	\$A7D7		; Get16BitPseudoRandomValueintoD()
4AB0: B1 0A AC	CMPA	\$0AAC		; Is it same random 8-bit as last time?
4AB3: 27 F6	BEQ	\$4AAB		;--/ If so, keep trying for different random number
				;
4AB5: B7 0A AC	STA	\$0AAC		; Save random number A into \$0AAC
				;
4AB8: A6 86	LDA	A,X		; A = RandomSoundEffectsAttractMode[A]
				;
4ABA: BD BF 1D	JSR	\$BF1D		; DoSoundTableIndexA(), plays random sound effect

```

4ABD: 25 12      BCS    $4AD1      ; if (ErrorPlayingSound)
                                   ; {
                                   ;   return;
                                   ; }
                                   ;
4ABF: 7C 0A AB    INC     $0AAB      ; A = $0AAB++
4AC2: B6 0A AB    LDA     $0AAB      ; if (A >= 3)  // if played 3 or more sound effects
4AC5: 81 03      CMPA   #$03        ; {
4AC7: 25 08      BCS     $4AD1      ;   Call $8C03, switch matrix related
4AC9: BD 8C 03    JSR     $8C03      ;   A--
4ACC: 00 F0      ;               ;   $36 += D
4ACE: 4A          DECA              ; }
4ACF: D3 36      ADDD    $36        ;
                                   ;
4AD1: 35 F6      PULS    A,B,X,Y,U,PC ; RTS
                                   ;
;-----;
                                   ;
4AD3: BD 86 CC    JSR     $86CC
4AD6: 46          RORA
4AD7: 50          NEGB
4AD8: 7F 0A AB    CLR     $0AAB
4ADB: 7E 99 28    JMP     $9928      ; ThreadedFunctionDone()
                                   ;
;-----;
                                   ;
; RandomSoundEffectsAttractMode[]
,
4ADE: 3E          ; SoundTableEntry3E, Sound6B, Monkey scream
4ADF: 3F          ; SoundTableEntry3F, Sound6C, Monkey scream
4AE0: 40          ; SoundTableEntry40, Sound6D, Monkey chirp
4AE1: 2D          ; SoundTableEntry2D, Sound6E, Burp
4AE2: 56          ; SoundTableEntry56, SoundA4, Airplane machine gun
4AE3: 4F          ; SoundTableEntry4F, Sound52, Gunshot Ricochet
4AE4: 50          ; SoundTableEntry50, Sound53, Gunshot
4AE5: 51          ; SoundTableEntry51, Sound54, Gunshot
4AE6: 7D          ; SoundTableEntry7D, Sound3D, Punch swish
4AE7: 7E          ; SoundTableEntry7E, Sound3E, Woosh Effect
4AE8: 78          ; SoundTableEntry78, Sound69, Thunderstrike
4AE9: 90          ; SoundTableEntry90, SoundAA, Thunder
4AEA: C8          ; SoundTableEntryC8, SoundDF, Arrows sound
4AEB: C9          ; SoundTableEntryC9, SoundF1, Arrow sound
                                   ;
;-----;
                                   ;

```

PlayRandomSoundEffect() (TS_LH6)

Below is the same function as how I found it in TS_LH6. I searched the ROM for byte patterns that I see in the IJ_L7 version of this function and eventually found this function. It is slightly different from the IJ_L7 because this uses 16-bit sound call indexes where IJ_L7 used 8-bit indexes for the attract mode SoundCallTable[] indexes. The code here wasn't annotated and has some alignment issues. The point is that by searching for this function you should be able to find a table of SoundCallTable[] indexes (assuming that all WPC games use this function).

```

;
;-----;-----
;
; PlayRandomSoundEffect ()
;
5B84: 34 76          PSHS  U,Y,X,B,A
5B86: 7D 04 71       TST   $0471
5B89: 26 74          BNE   $5BFF
5B8B: 0D 8D          TST   $008D
5B8D: 27 70          BEQ   $5BFF
5B8F: BD 87 09       JSR   $8709
5B92: 9A 01          ORA   $0001
5B94: 25 0B          BCS   $5BA1
5B96: 86 37          LDA   #$37
5B98: B7 0A B6       STA   $0AB6
5B9B: BD 86 01       JSR   $8601
5B9E: 04 20          LSR   $0020
5BA0: 5E           Illegal Opcode
5BA1: BD 87 51       JSR   $8751
5BA4: 01           Illegal Opcode
5BA5: 1E 24          EXG   Y,S
5BA7: 4F           CLRA
5BA8: BD 9C 1C       JSR   $9C1C
5BAB: 10 8C 00 89    CMPY  #$0089
5BAF: 27 4E          BEQ   $5BFF
5BB1: BD 87 09       JSR   $8709
5BB4: 04 01          LSR   $0001
5BB6: 25 47          BCS   $5BFF
5BB8: B6 0B 6B       LDA   $0B6B
5BBB: 26 42          BNE   $5BFF
5BBD: BD CB 28       JSR   $CB28
5BC0: 81 F3          CMPA  #$F3
5BC2: 27 3B          BEQ   $5BFF
5BC4: 86 3A          LDA   #$3A
```



```

5BC6: BD CB 50      JSR    $CB50
5BC9: 24 34          BCC    $5BFF
5BCB: B6 1D 5A      LDA    $1D5A
5BCE: BD 89 B7      JSR    $89B7
5BD1: 61            Illegal Opcode
5BD2: 74 35 8E      LSR    $358E
5BD5: 5C            INCB
5BD6: 0C 86          INC     $0086
5BD8: 17 BD A6      LBSR   $1981
5BDB: FC B1 0A      LDD     $B10A
5BDE: B5 27 F6      BITA   $27F6
5BE1: B7 0A B5      STA    $0AB5
5BE4: 48            ASLA
5BE5: 10 AE 86      LDY     A,X
5BE8: BD BE 99      JSR    $BE99
5BEB: 25 12          BCS    $5BFF
5BED: 7C 0A B4      INC     $0AB4
5BF0: B6 0A B4      LDA    $0AB4
5BF3: 81 03          CMPA   #$03
5BF5: 25 08          BCS    $5BFF
5BF7: BD 8C 72      JSR    $8C72
5BFA: 01            Illegal Opcode
5BFB: 1E 5C          EXG     PC,inv
5BFD: 01            Illegal Opcode
5BFE: 34 35          PSHS   Y,X,B,CC
5C00: F6 BD 87      LDB     $BD87
5C03: 27 46          BEQ     $5C4B
5C05: 50            NEGB
5C06: 7F 0A B4      CLR     $0AB4
5C09: 7E 99 A3      JMP     $99A3
;
;-----;
;
; RandomSoundEffectsAttractMode[]
,
5C0C: 00 F8          ; SoundTableEntryF8, Gunshot
5C0E: 00 E6          ; SoundTableEntryE6, Ooogh
5C10: 00 DB          ; SoundTableEntryDB, smashy
5C12: 00 D6          ; SoundTableEntryD6, action sound
5C14: 01 37          ; SoundTableEntry137, Who's That?
5C16: 01 67          ; SoundTableEntry167, doppler
5C18: 00 71          ; SoundTableEntry71, action effect
5C1A: 00 72          ; SoundTableEntry72, action effect
5C1C: 00 2B          ; SoundTableEntry2B, "Ow!"
5C1E: 00 1E          ; SoundTableEntry1E, door slam

```

5C20: 00 1A	; SoundTableEntry1A, woush effect
5C22: 01 8D	; SoundTableEntry18D, award, horns
5C24: 00 E6	; SoundTableEntryE6, Ooogh
5C26: 00 72	; SoundTableEntry72, action effect
5C28: 00 75	; SoundTableEntry75, sound effect
5C2A: 00 77	; SoundTableEntry77, smash effect
5C2C: 00 97	; SoundTableEntry97, explosion
5C2E: 00 A2	; SoundTableEntryA2, cymbol effect
5C30: 00 A5	; SoundTableEntryA5, laser sound
5C32: 01 98	; SoundTableEntry198, Moooh!
5C34: 01 9A	; SoundTableEntry19A, quack!
5C36: 00 F3	; SoundTableEntryF3, smack effect
5C38: 01 2F	; SoundTableEntry12F, sound effect
	;
;-----;	

Using the information in this document

If you're interested in playing with the sound calls in a particular ROM I'd suggest using the info in this document to identify in your ROM the SoundCallTable[] as well as the various functions that I mentioned in this document. Next I'd suggest mapping out all of the entries in your SoundCallTable[] and label each entry with its 0-based index along with description of the sound. You can use PinMame with F4 sound mode to dial in each sound call number to see what they do. This may take awhile but it will make future sound call hacks on your game easier. Remember not to confuse a SoundCallTable[] index with the actual sound call number which is the number sent to the sound card (and the number you want to dial into PinMame F4 mode).

Next you can play with changing sounds around, or if you're very adventurous you can even insert new sound calls. By finding certain places in the code where you want to add a sound call you could jump to unused portion of code (in the same bank) and make a call to one of the functions shown above to queue up a sound call, then to back to normal code. This is definitely a more advanced sort of ROM hack but I can help if you need any assistance in this area.

Another idea is that you can make a SoundCallTable[] entry point to a new list of Sound Commands, located in unused region of the same bank, and there you can play with adding a series of sound calls in place of the single sound call you're replacing.

-garrett lee, mrglee@yahoo.com