Instruction Name	Length	Example / Info	History

{00} Nop	01	00++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x00`
		`} Nop;`
		This bytecode is used for alignment of 1-byte opcodes and ending ElseIf blocks.
			08-02-2024 Newly Added

{01} Evt_end	02	01 00++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x01`
		`  UCHAR zAlign;  // Always Zero (Alignment byte)`
		`} Evt_end;`
		This bytecode ends the current Main/Sub script.
			08-02-2024 Newly Added

{02} Evt_next	01	02++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x02`
		`} Evt_next;`
		This bytecode moves to the next event in the sequence.
			08-02-2024 Newly Added

{03} Evt_chain	02	03 ID++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x03`
		`  UCHAR NextEventId;  // Event ID to chain to`
		`} Evt_chain;`
		This bytecode chains the current event to the specified next event ID, allowing the script to continue execution from the linked event.
			08-02-2024 Newly Added

{04} Evt_exec	04	04 ?? ?? ID++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x04`
		`  UCHAR data1;  // Typically FF`
		`  UCHAR GoSub;  // Opcode for GoSub 0x18`
		`  UCHAR ScdId;  // Sub Script ID to Jump to`
		`} Evt_exec;`
		This bytecode executes the specified event with given parameters.
			08-02-2024 Newly Added

{05} Evt_kill	02	05 ID++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x05`
		`  UCHAR EventId;  // Event ID to terminate`
		`} Evt_kill;`
		This bytecode terminates the specified event.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{06} Ifel_ck	04	06 00 SI ZE++
		typedef struct {
		UCHAR Opcode; // 0x06
		UCHAR zAlign; // Always Zero (Alignment byte)
		USHORT Size; // Size of the block to check
		} Ifel_ck;
		This bytecode checks a condition and branches accordingly.
			08-02-2024 Newly Added

{07} Else_ck	04	07 00 SI ZE++
		typedef struct {
		UCHAR Opcode; // 0x07
		UCHAR zAlign; // Always Zero (Alignment byte)
		USHORT Size; // Size of the block to check
		} Else_ck;
		This bytecode specifies the size of the block to check if the corresponding Ifel_ck condition is met.
			08-02-2024 Newly Added

{08} Endif	02	08 00++
		typedef struct {
		UCHAR Opcode; // 0x08
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Endif;
		This bytecode marks the end of an If/ElseIf/Else block.
			08-02-2024 Newly Added

{09} Sleep	04	09 ?? CO NT++
		typedef struct {
		UCHAR Opcode; // 0x09
		UCHAR Sleeping; // Opcode for Sleeping 0x0A
		USHORT Count; // Timer / Sleep Duration
		} Sleep;
		This bytecode pauses script execution for the specified duration.
			08-02-2024 Newly Added

{0A} Sleeping	03	0A CO NT++
		typedef struct {
		UCHAR Opcode; // 0x0A
		USHORT Count; // Timer / Sleep Duration
		} Sleeping;
		This bytecode pauses script execution for the specified duration.
			08-02-2024 Newly Added

{0B} Wsleep	01	0B++
		typedef struct {
		UCHAR Opcode; // 0x0B
		} Wsleep;
		This bytecode used before 0C will wait until the current XA sound has finished playing before proceeding.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{0C} Wsleeping	01	0C++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x0C`
		`} Wsleeping;`
		This bytecode used after 0B will wait until the current XA sound has finished playing before proceeding.
			08-02-2024 Newly Added

{0D} For	06	0D 00 SI ZE CO NT++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x0D`
		`  UCHAR zAlign;  // Always Zero (Alignment byte)`
		`  USHORT Size;  // Size of the block to check`
		`  USHORT Count;  // Amount of times block is looped`
		`} For;`
		This bytecode begins a for-loop with the specified start and end values.
			08-02-2024 Newly Added

{0E} Next	02	0E 00++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x0E`
		`  UCHAR zAlign;  // Always Zero (Alignment byte)`
		`} Next;`
		This bytecode marks the end of a for-loop.
			08-02-2024 Newly Added

{0F} While	04	0F 00 SI ZE++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x0F`
		`  UCHAR zAlign;  // Always Zero (Alignment byte)`
		`  USHORT Size;  // Size of the block to check`
		`} While;`
		This bytecode begins a while-loop that continues as long as the specified condition is true.
			08-02-2024 Newly Added

{10} Ewhile	02	10 ID++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x10`
		`  UCHAR LoopId;  // ID of the while-loop to end`
		`} Ewhile;`
		This bytecode ends the specified while-loop.
			08-02-2024 Newly Added

{11} Do	04	11 00 SI ZE++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x11`
		`  UCHAR zAlign;  // Always Zero (Alignment byte)`
		`  USHORT Size;  // Size of the block to check`
		`} Do;`
		This bytecode begins a do-while loop that executes the loop body once before checking the condition.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{12} Edwhile	02	12 ID++
		typedef struct {
		UCHAR Opcode; // 0x12
		UCHAR LoopId; // ID of the do-while loop to end
		} Edwhile;
		This bytecode ends the specified do-while loop.
			08-02-2024 Newly Added

{13} Switch	04	13 ID SI ZE++
		typedef struct {
		UCHAR Opcode; // 0x13
		UCHAR SwitchId; // ID of the switch variable
		USHORT Size; // Size of the block to check
		} Switch;
		This bytecode begins a switch-case block with the specified switch variable and default size.
			08-02-2024 Newly Added

{14} Case	06	14 00 SI ZE VA LU++
		typedef struct {
		UCHAR Opcode; // 0x14
		UCHAR zAlign; // Always Zero (Alignment byte)
		USHORT Size; // Size of the block to check
		USHORT CaseValue; // Value to compare with the switch variable
		} Case;
		This bytecode defines a case within a switch-case block.
			08-02-2024 Newly Added

{15} Default	02	15 00++
		typedef struct {
		UCHAR Opcode; // 0x15
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Default;
		This bytecode marks the default case in a switch-case block.
			08-02-2024 Newly Added

{16} Eswitch	02	16 00++
		typedef struct {
		UCHAR Opcode; // 0x16
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Eswitch;
		This bytecode ends the switch-case block.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{17} Goto	06	17 ?? ?? 00 XX XX++
		typedef struct {
		UCHAR Opcode; // 0x17
		UCHAR Ifel_ctr; // Always 0xFF (0x01 on r304-sub05, only)
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT Offset; // Relative Pointer, always references same script
		} Goto;
		This bytecode jumps to the specified offset within the script.
			08-02-2024 Newly Added

{18} Gosub	02	18 ID++
		typedef struct {
		UCHAR Opcode; // 0x18
		UCHAR SubroutineId; // ID of the subroutine to call
		} Gosub;
		This bytecode calls the specified subroutine.
			08-02-2024 Newly Added

{19} Return	02	19 00++
		typedef struct {
		UCHAR Opcode; // 0x19
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Return;
		This bytecode returns from the specified subroutine.
			08-02-2024 Newly Added

{1A} Break	02	1A 00++
		typedef struct {
		UCHAR Opcode; // 0x1A
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Break;
		This bytecode breaks out of the specified loop.
			08-02-2024 Newly Added

{1B} For2	06	1B 00 XX XX 00 XX XX++
		typedef struct {
		UCHAR Opcode; // 0x1B
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT StartValue; // Start value of the loop counter
		UCHAR zAlign2; // Always Zero (Alignment byte)
		SHORT EndValue; // End value of the loop counter
		} For2;
		This bytecode begins a for-loop with the specified start and end values.
			08-02-2024 Newly Added

{1C} Break_point	01	1C++
		typedef struct {
		UCHAR Opcode; // 0x1C
		} Break_point;
		This bytecode sets a breakpoint for debugging purposes.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{1D} Work_copy	04	1D ?? ID ??++
		typedef struct {
		UCHAR Opcode; // 0x1D
		UCHAR Source; // Source index
		UCHAR DestinationId; // Destination index
		UCHAR Typecast; // Typecast operation
		} Work_copy;
		This bytecode copies a value from the source index to the destination index with an optional typecast.
			08-02-2024 Newly Added

{1E} Nop	01	1E++
		typedef struct {
		UCHAR Opcode; // 0x1E
		} Nop;
		This bytecode is used for alignment of 1-byte opcodes and ending ElseIf blocks.
			08-02-2024 Newly Added

{1F} Nop	01	1F++
		typedef struct {
		UCHAR Opcode; // 0x1F
		} Nop;
		This bytecode is used for alignment of 1-byte opcodes and ending ElseIf blocks.
			08-02-2024 Newly Added

{20} Nop	01	20++
		typedef struct {
		UCHAR Opcode; // 0x20
		} Nop;
		This bytecode is used for alignment of 1-byte opcodes and ending ElseIf blocks.
			08-02-2024 Newly Added

{21} Ck	04	21 ?? ID ??++
		typedef struct {
		UCHAR Opcode; // 0x21
		UCHAR Flag; // System flag to check
		UCHAR Id; // Bit ID to check
		UCHAR OnOff; // On/Off state to check
		} Ck;
		This bytecode checks the specified system flag and bit ID for the given On/Off state.
			08-02-2024 Newly Added

{22} Set	04	22 ?? ID ??++
		typedef struct {
		UCHAR Opcode; // 0x22
		UCHAR Flag; // System flag to set
		UCHAR Id; // Bit ID to set
		UCHAR OnOff; // On/Off state to set
		} Set;
		This bytecode sets the specified system flag and bit ID to the given On/Off state.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{23} Cmp	06	23 ?? ?? VA LU++
		typedef struct {
		UCHAR Opcode; // 0x23
		UCHAR Flag; // System flag to compare
		UCHAR Operator; // Comparison operator
		USHORT Value; // Value to compare against
		} Cmp;
		This bytecode compares the specified system flag with the given value using the provided comparison operator.
			08-02-2024 Newly Added

{24} Save	04	24 ID XX XX++
		typedef struct {
		UCHAR Opcode; // 0x24
		UCHAR Destination; // Destination index
		SHORT Source; // Source value
		} Save;
		This bytecode saves the specified source value to the destination index.
			08-02-2024 Newly Added

{25} Copy	03	25 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x25
		UCHAR Destination; // Destination index
		UCHAR Source; // Source index
		} Copy;
		This bytecode copies the value from the source index to the destination index.
			08-02-2024 Newly Added

{26} Calc	06	26 00 ?? ?? VA LU++
		typedef struct {
		UCHAR Opcode; // 0x26
		UCHAR zAlign; // Always Zero (Alignment byte)
		UCHAR Operator; // Arithmetic operation to perform
		UCHAR Flag; // Memory Location to apply math to
		SHORT Value; // Amount used in operation
		} Calc;
		This bytecode performs the specified arithmetic operation on the operands and stores the result.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{27} Calc2	04	27 ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x27
		UCHAR Operand1; // First operand
		UCHAR Operator; // Arithmetic operation to perform
		UCHAR Operand2; // Second operand
		} Calc2;
		This bytecode performs the specified arithmetic operation on the two operands and stores the result.
			08-02-2024 Newly Added

{28} Sce_rnd	01	28++
		typedef struct {
		UCHAR Opcode; // 0x28
		} Sce_rnd;
		This bytecode generates a random value.
			08-02-2024 Newly Added

{29} Cut_chg	02	29 ID++
		typedef struct {
		UCHAR Opcode; // 0x29
		UCHAR CutId; // ID of the cutscene
		} Cut_chg;
		This bytecode changes the current camera to the specified camera ID.
			08-02-2024 Newly Added

{2A} Cut_old	01	2A++
		typedef struct {
		UCHAR Opcode; // 0x2A
		} Cut_old;
		This bytecode reverts to the previous camera.
			08-02-2024 Newly Added

{2B} Message_on	06	2B ?? ID 00 SI ZE++
		typedef struct {
		UCHAR Opcode; // 0x2B
		UCHAR Type; // Message type
		UCHAR MessageId; // ID of the message to display
		UCHAR zAlign; // Always Zero (Alignment byte)
		USHORT DisplayTime; // Time to display the message
		} Message_on;
		This bytecode displays the specified message.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{2C} Aot_set	20	2C ID ?? ?? FL ?? XX XX ZZ ZZ WW WW DD DD ?? ?? ?? ?? ?? ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x2C`
		`  UCHAR AotId;  // AOT ID`
		`  UCHAR zAlign0;  // Always Zero (Alignment byte)`
		`  UCHAR Floor;  // Floor number`
		`  UCHAR Flag;  // Flag`
		`  SHORT X;  // X position`
		`  SHORT Z;  // Z position`
		`  USHORT Width;  // Width`
		`  USHORT Depth;  // Depth`
		`  UCHAR Data0;  // Data 0`
		`  UCHAR Data1;  // Data 1`
		`  USHORT Data2;  // Data 2`
		`  USHORT Data3;  // Data 3`
		`  USHORT Data4;  // Data 4`
		`} Aot_set;`
		This bytecode sets the properties of the specified AOT.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{2D} Obj_model_set	38	2D ??++
		`typedef struct {`
		`    UCHAR Opcode;        // 0x2D`
		`    UCHAR ModelId;        // Model ID`
		`    UCHAR zAlign0;        // Always Zero (Alignment byte)`
		`    UCHAR Floor;        // Floor number`
		`    UCHAR zAlign1;        // Always Zero (Alignment byte)`
		`    UCHAR zAlign2;        // Always Zero (Alignment byte)`
		`    USHORT Type;        // Object type`
		`    USHORT Flag;        // Flag`
		`    SHORT X;        // X position`
		`    SHORT Y;        // Y position`
		`    SHORT Z;        // Z position`
		`    SHORT DirX;        // X direction`
		`    SHORT DirY;        // Y direction`
		`    SHORT DirZ;        // Z direction`
		`    SHORT OffsetX;        // Offset X`
		`    SHORT OffsetY;        // Offset Y`
		`    SHORT OffsetZ;        // Offset Z`
		`    SHORT SizeX;        // Size X`
		`    SHORT SizeY;        // Size Y`
		`    SHORT SizeZ;        // Size Z`
		`    UCHAR Data0;        // Data 0`
		`    UCHAR Data1;        // Data 1`
		`    USHORT Data2;        // Data 2`
		`    USHORT Data3;        // Data 3`
		`    USHORT Data4;        // Data 4`
		`    USHORT Data5;        // Data 5`
		`    USHORT Data6;        // Data 6`
		`    USHORT Data7;        // Data 7`
		`    USHORT Data8;        // Data 8`
		`    USHORT Data9;        // Data 9`
		`} Obj_model_set;`
		This bytecode sets the properties of the specified object model.
			08-02-2024 Newly Added

{2E} Work_set	03	2E ?? ID++
		`typedef struct {`
		`    UCHAR Opcode;        // 0x2E`
		`    UCHAR Type;        // Type of Work Set to Select`
		`    UCHAR EntityId;        // ID of Entity to select`
		`} Work_set;`
		This bytecode sets the properties of the specified work (task).
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{2F} Speed_set	04	2F ID ?? 00++
		typedef struct {
		UCHAR Opcode; // 0x2F
		UCHAR SpeedId; // ID of the speed setting
		USHORT SpeedValue; // Value of the speed setting
		} Speed_set;
		This bytecode sets the specified speed setting.
			08-02-2024 Newly Added

{30} Add_speed	01	30++
		typedef struct {
		UCHAR Opcode; // 0x30
		} Add_speed;
		This bytecode increments the speed setting.
			08-02-2024 Newly Added

{31} Add_aspeed	01	31++
		typedef struct {
		UCHAR Opcode; // 0x31
		} Add_aspeed;
		This bytecode increments the angular speed setting.
			08-02-2024 Newly Added

{32} Pos_set	08	32 00 XX XX YY YY ZZ ZZ++
		typedef struct {
		UCHAR Opcode; // 0x32
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		} Pos_set;
		This bytecode sets the position in 3D space.
			08-02-2024 Newly Added

{33} Dir_set	08	33 00 DX DX DY DY DZ DZ++
		typedef struct {
		UCHAR Opcode; // 0x33
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		} Dir_set;
		This bytecode sets the direction in 3D space.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{34} Member_set	04	34 ID ?? 00++
		typedef struct {
		UCHAR Opcode; // 0x34
		UCHAR MemberId; // ID of the member
		USHORT Value; // Value to set
		} Member_set;
		This bytecode sets the properties of the specified member.
			08-02-2024 Newly Added

{35} Member_set2	03	35 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x35
		UCHAR MemberId; // ID of the member
		UCHAR Value; // Value to set
		} Member_set2;
		This bytecode sets a single property of the specified member.
			08-02-2024 Newly Added

{36} Se_on	12	36 ID ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x36
		UCHAR SeId; // ID of the sound effect to play
		UCHAR Volume; // Volume of the sound effect
		UCHAR Pan; // Pan of the sound effect
		UCHAR Param1; // Parameter 1 for the sound effect
		UCHAR Param2; // Parameter 2 for the sound effect
		UCHAR Param3; // Parameter 3 for the sound effect
		UCHAR Param4; // Parameter 4 for the sound effect
		UCHAR Param5; // Parameter 5 for the sound effect
		UCHAR Param6; // Parameter 6 for the sound effect
		UCHAR Param7; // Parameter 7 for the sound effect
		UCHAR Param8; // Parameter 8 for the sound effect
		} Se_on;
		This bytecode plays the specified sound effect with the given parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{37} Sca_id_set	04	37 ID ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x37
		UCHAR ScaId; // ID of the scale to set
		UCHAR Value; // Value to set the scale to
		UCHAR Type; // Type of scale variable
		} Sca_id_set;
		This bytecode sets the specified scale ID to the given value.
			08-02-2024 Newly Added

{38} Flr_set	03	38 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x38
		UCHAR FlrId; // ID of the floor to set
		UCHAR Value; // Value to set the floor to
		} Flr_set;
		This bytecode sets the specified floor ID to the given value.
			08-02-2024 Newly Added

{39} Dir_ck	08	39 00 DX DX DY DY DZ DZ++
		typedef struct {
		UCHAR Opcode; // 0x39
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		} Dir_ck;
		This bytecode checks the specified direction coordinates.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{3A} Sce_espr_on	16	3A ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x3A
		UCHAR EsprId; // ID of the espr to turn on
		UCHAR Param1; // Parameter 1
		UCHAR Param2; // Parameter 2
		UCHAR Param3; // Parameter 3
		UCHAR Param4; // Parameter 4
		UCHAR Param5; // Parameter 5
		UCHAR Param6; // Parameter 6
		UCHAR Param7; // Parameter 7
		UCHAR Param8; // Parameter 8
		UCHAR Param9; // Parameter 9
		UCHAR Param10; // Parameter 10
		UCHAR Param11; // Parameter 11
		UCHAR Param12; // Parameter 12
		UCHAR Param13; // Parameter 13
		UCHAR Param14; // Parameter 14
		UCHAR Param15; // Parameter 15
		} Sce_espr_on;
		This bytecode turns on the specified espr with the given parameters.
			08-02-2024 Newly Added

<table>
<tr><th>Instruction Name</th><th>Length</th><th>Example / Info</th><th>History</th></tr>
<tr><td>{3B} Door_aot_set</td><td>32</td><td>3B ??++
<pre>
typedef struct {
    UCHAR Opcode;   // 0x3B
    UCHAR AotType;   // Type of AOT to set
    UCHAR Id;   // ID of the AOT to set
    UCHAR Floor;   // Floor number
    UCHAR Super;   // Super parameter
    UCHAR X;   // X coordinate
    UCHAR Y;   // Y coordinate
    UCHAR Z;   // Z coordinate
    UCHAR Width;   // Width of the AOT
    UCHAR Depth;   // Depth of the AOT
    UCHAR Height;   // Height of the AOT
    UCHAR DirX;   // X direction
    UCHAR DirY;   // Y direction
    UCHAR DirZ;   // Z direction
    UCHAR InitParam;   // Initial parameter
    UCHAR EventParam;   // Event parameter
    UCHAR ConfigParam;   // Configuration parameter
    UCHAR Type;   // Type of the AOT
    UCHAR Flags;   // Flags for the AOT
    UCHAR Reserved;   // Reserved byte
} Door_aot_set;
</pre>
This bytecode sets the specified door AOT parameters.</td><td>08-02-2024 Newly Added</td></tr>
<tr><td>{3C} Cut_auto</td><td>02</td><td>3C ID++
<pre>
typedef struct {
    UCHAR Opcode;   // 0x3C
    UCHAR AutoId;   // ID of the auto cutscene to set
} Cut_auto;
</pre>
This bytecode sets the specified auto cutscene parameters.</td><td>08-02-2024 Newly Added</td></tr>
</table>

Instruction Name	Length	Example / Info	History

{3D} Member_copy	03	3D ID ??++
		typedef struct {
		UCHAR Opcode; // 0x3D
		UCHAR SourceId; // ID of the source member to copy from
		UCHAR DestinationId; // ID of the destination member to copy to
		} Member_copy;
		This bytecode copies the specified member parameters from the source to the destination.
			08-02-2024 Newly Added

{3E} Member_cmp	06	3E 00 ?? ?? VA LU++
		typedef struct {
		UCHAR Opcode; // 0x3E
		UCHAR zAlign; // Always Zero (Alignment byte)
		UCHAR Flag; // Load_member_addr_branch() argv[1]
		UCHAR Operator; // Comparison operator
		SHORT Value; // Value to compare
		} Member_cmp;
		This bytecode compares the specified member parameters with the given value.
			08-02-2024 Newly Added

{3F} Plc_motion	04	3F ID ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x3F
		UCHAR MotionId; // ID of the motion to set
		UCHAR Mode; // Mode to set the motion to
		UCHAR Param; // Parameter for the motion
		} Plc_motion;
		This bytecode sets the specified motion parameters.
			08-02-2024 Newly Added

{40} Plc_dest	08	40 00 ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x40
		UCHAR zAlign; // Always Zero (Alignment byte)
		UCHAR DestId; // ID of the destination to set
		UCHAR Mode; // Mode to set the destination to
		UCHAR Param1; // Parameter 1 for the destination
		UCHAR Param2; // Parameter 2 for the destination
		UCHAR Param3; // Parameter 3 for the destination
		} Plc_dest;
		This bytecode sets the specified destination parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{41} Plc_neck	10	41 00 ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x41
		UCHAR zAlign; // Always Zero (Alignment byte)
		UCHAR NeckId; // ID of the neck to set
		UCHAR Param1; // Parameter 1 for the neck
		UCHAR Param2; // Parameter 2 for the neck
		UCHAR Param3; // Parameter 3 for the neck
		UCHAR Param4; // Parameter 4 for the neck
		UCHAR Param5; // Parameter 5 for the neck
		UCHAR Param6; // Parameter 6 for the neck
		UCHAR Param7; // Parameter 7 for the neck
		} Plc_neck;
		This bytecode sets the specified neck parameters.
			08-02-2024 Newly Added

{42} Plc_ret	01	42++
		typedef struct {
		UCHAR Opcode; // 0x42
		} Plc_ret;
		This bytecode returns from the specified plc function.
			08-02-2024 Newly Added

{43} Plc_flg	04	43 ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x43
		UCHAR Type; // Type of the flag
		USHORT Flag; // Flag value
		} Plc_flg;
		This bytecode sets the specified flag parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{44} Sce_em_set	22	44 ??++
		typedef struct {
		UCHAR Opcode; // 0x44
		UCHAR EmId; // ID of the enemy to set
		UCHAR Param1; // Parameter 1
		UCHAR Param2; // Parameter 2
		UCHAR Param3; // Parameter 3
		UCHAR Param4; // Parameter 4
		UCHAR Param5; // Parameter 5
		UCHAR Param6; // Parameter 6
		UCHAR Param7; // Parameter 7
		UCHAR Param8; // Parameter 8
		UCHAR Param9; // Parameter 9
		UCHAR Param10; // Parameter 10
		UCHAR Param11; // Parameter 11
		UCHAR Param12; // Parameter 12
		UCHAR Param13; // Parameter 13
		UCHAR Param14; // Parameter 14
		UCHAR Param15; // Parameter 15
		UCHAR Param16; // Parameter 16
		UCHAR Param17; // Parameter 17
		UCHAR Param18; // Parameter 18
		UCHAR Param19; // Parameter 19
		UCHAR Param20; // Parameter 20
		UCHAR Param21; // Parameter 21
		} Sce_em_set;
		This bytecode sets the specified enemy parameters.
			08-02-2024 Newly Added

{45} Col_chg_set	05	45 ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x45
		UCHAR ColId; // ID of the color to set
		UCHAR Value; // Value to set the color to
		UCHAR Type; // Type of color operation
		} Col_chg_set;
		This bytecode sets the specified color parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{46} Aot_reset	10	46 ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x46
		UCHAR AotType; // Type of AOT to reset
		UCHAR Id; // ID of the AOT to reset
		UCHAR Param1; // Parameter 1
		UCHAR Param2; // Parameter 2
		UCHAR Param3; // Parameter 3
		UCHAR Param4; // Parameter 4
		UCHAR Param5; // Parameter 5
		UCHAR Param6; // Parameter 6
		UCHAR Param7; // Parameter 7
		} Aot_reset;
		This bytecode resets the specified AOT parameters.
			08-02-2024 Newly Added

{47} Aot_on	02	47 00++
		typedef struct {
		UCHAR Opcode; // 0x47
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Aot_on;
		This bytecode turns on the specified AOT.
			08-02-2024 Newly Added

{48} Super_set	16	48 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x48
		UCHAR SuperId; // ID of the super parameter to set
		UCHAR Param1; // Parameter 1
		UCHAR Param2; // Parameter 2
		UCHAR Param3; // Parameter 3
		UCHAR Param4; // Parameter 4
		UCHAR Param5; // Parameter 5
		UCHAR Param6; // Parameter 6
		UCHAR Param7; // Parameter 7
		UCHAR Param8; // Parameter 8
		UCHAR Param9; // Parameter 9
		UCHAR Param10; // Parameter 10
		UCHAR Param11; // Parameter 11
		UCHAR Param12; // Parameter 12
		UCHAR Param13; // Parameter 13
		UCHAR Param14; // Parameter 14
		UCHAR Param15; // Parameter 15
		} Super_set;
		This bytecode sets the specified super parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{49} Super_reset	08	49 ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x49
		UCHAR SuperId; // ID of the super parameter to reset
		UCHAR Param1; // Parameter 1
		UCHAR Param2; // Parameter 2
		UCHAR Param3; // Parameter 3
		UCHAR Param4; // Parameter 4
		UCHAR Param5; // Parameter 5
		} Super_reset;
		This bytecode resets the specified super parameters.
			08-02-2024 Newly Added

{4A} Plc_gun	02	4A ID++
		typedef struct {
		UCHAR Opcode; // 0x4A
		UCHAR GunId; // ID of the gun to set
		} Plc_gun;
		This bytecode sets the specified gun parameters.
			08-02-2024 Newly Added

{4B} Cut_replace	03	4B ID ??++
		typedef struct {
		UCHAR Opcode; // 0x4B
		UCHAR CutId; // ID of the cutscene to replace
		UCHAR Param; // Parameter for the cutscene
		} Cut_replace;
		This bytecode replaces the specified cutscene parameters.
			08-02-2024 Newly Added

{4C} Sce_espr_kill	05	4C ID ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x4C
		UCHAR EsprId; // ID of the espr to kill
		UCHAR Mode; // Mode to set the espr to
		UCHAR Param1; // Parameter 1 for the espr
		UCHAR Param2; // Parameter 2 for the espr
		} Sce_espr_kill;
		This bytecode kills the specified espr with the given parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{4D} Door_model_set	22	4D ID XX XX YY YY ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x4D`
		`  UCHAR ModelId;  // ID of the door model to set`
		`  SHORT PosX;  // X position`
		`  SHORT PosY;  // Y position`
		`  USHORT Param1;  // Parameter 1 for the door model`
		`  USHORT Param2;  // Parameter 2 for the door model`
		`  USHORT Param3;  // Parameter 3 for the door model`
		`  USHORT Param4;  // Parameter 4 for the door model`
		`  USHORT Param5;  // Parameter 5 for the door model`
		`  USHORT Param6;  // Parameter 6 for the door model`
		`  USHORT Param7;  // Parameter 7 for the door model`
		`  USHORT Param8;  // Parameter 8 for the door model`
		`  USHORT Param9;  // Parameter 9 for the door model`
		`  USHORT Param10;  // Parameter 10 for the door model`
		`} Door_model_set;`
		This bytecode sets the specified door model parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{4E} Item_aot_set	22	4E ??++
		typedef struct {
		UCHAR Opcode; // 0x4E
		UCHAR AotType; // Type of AOT to set
		UCHAR Id; // ID of the AOT to set
		UCHAR Floor; // Floor number
		UCHAR Super; // Super parameter
		UCHAR X; // X coordinate
		UCHAR Y; // Y coordinate
		UCHAR Z; // Z coordinate
		UCHAR Width; // Width of the AOT
		UCHAR Depth; // Depth of the AOT
		UCHAR Height; // Height of the AOT
		UCHAR DirX; // X direction
		UCHAR DirY; // Y direction
		UCHAR DirZ; // Z direction
		UCHAR InitParam; // Initial parameter
		UCHAR EventParam; // Event parameter
		UCHAR ConfigParam; // Configuration parameter
		UCHAR Type; // Type of the AOT
		UCHAR Flags; // Flags for the AOT
		UCHAR Reserved; // Reserved byte
		} Item_aot_set;
		This bytecode sets the specified item AOT parameters.
			08-02-2024 Newly Added

{4F} Sce_key_ck	04	4F ID ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x4F
		UCHAR KeyId; // ID of the key to check
		UCHAR Mode; // Mode to set the key to
		UCHAR Param; // Parameter for the key
		} Sce_key_ck;
		This bytecode checks the specified key parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{50} Sce_trg_ck	04	50 ID ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x50
		UCHAR TrgId; // ID of the trigger to check
		UCHAR Mode; // Mode to set the trigger to
		UCHAR Param; // Parameter for the trigger
		} Sce_trg_ck;
		This bytecode checks the specified trigger parameters.
			08-02-2024 Newly Added

{51} Sce_bgm_control	06	51 ID ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x51
		UCHAR BgmId; // ID of the BGM to control
		UCHAR Mode; // Mode to set the BGM to
		UCHAR Param1; // Parameter 1 for the BGM
		UCHAR Param2; // Parameter 2 for the BGM
		UCHAR Param3; // Parameter 3 for the BGM
		} Sce_bgm_control;
		This bytecode controls the specified BGM parameters.
			08-02-2024 Newly Added

{52} Sce_espr_control	06	52 ID ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x52
		UCHAR EsprId; // ID of the espr to control
		UCHAR Mode; // Mode to set the espr to
		UCHAR Param1; // Parameter 1 for the espr
		UCHAR Param2; // Parameter 2 for the espr
		UCHAR Param3; // Parameter 3 for the espr
		} Sce_espr_control;
		This bytecode controls the specified espr parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{53} Sce_fade_set	06	53 ID ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x53
		UCHAR FadeId; // ID of the fade to set
		UCHAR Mode; // Mode to set the fade to
		UCHAR Param1; // Parameter 1 for the fade
		UCHAR Param2; // Parameter 2 for the fade
		UCHAR Param3; // Parameter 3 for the fade
		} Sce_fade_set;
		This bytecode sets the specified fade parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{54} Sce_espr3d_on	22	54 ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x54`
		`  UCHAR Espr3dId;  // ID of the 3D espr to turn on`
		`  UCHAR Param1;  // Parameter 1`
		`  UCHAR Param2;  // Parameter 2`
		`  UCHAR Param3;  // Parameter 3`
		`  UCHAR Param4;  // Parameter 4`
		`  UCHAR Param5;  // Parameter 5`
		`  UCHAR Param6;  // Parameter 6`
		`  UCHAR Param7;  // Parameter 7`
		`  UCHAR Param8;  // Parameter 8`
		`  UCHAR Param9;  // Parameter 9`
		`  UCHAR Param10;  // Parameter 10`
		`  UCHAR Param11;  // Parameter 11`
		`  UCHAR Param12;  // Parameter 12`
		`  UCHAR Param13;  // Parameter 13`
		`  UCHAR Param14;  // Parameter 14`
		`  UCHAR Param15;  // Parameter 15`
		`  UCHAR Param16;  // Parameter 16`
		`  UCHAR Param17;  // Parameter 17`
		`  UCHAR Param18;  // Parameter 18`
		`  UCHAR Param19;  // Parameter 19`
		`  UCHAR Param20;  // Parameter 20`
		`  UCHAR Param21;  // Parameter 21`
		`} Sce_espr3d_on;`
		This bytecode turns on the specified 3D espr with the given parameters.
			08-02-2024 Newly Added

{55} Member_calc	06	55 ID ?? ?? VA LU++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x55`
		`  UCHAR MemberId;  // ID of the member to calculate`
		`  UCHAR Operator;  // Arithmetic operation to perform`
		`  USHORT Value;  // Value to use in the calculation`
		`} Member_calc;`
		This bytecode performs the specified arithmetic operation on the member with the given value.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{56} Member_calc2	04	56 ID ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x56
		UCHAR MemberId; // ID of the member to calculate
		UCHAR Operator; // Arithmetic operation to perform
		UCHAR Value; // Value to use in the calculation
		} Member_calc2;
		This bytecode performs the specified arithmetic operation on the member with the given value.
			08-02-2024 Newly Added

{57} Sce_bgmtbl_set	08	57 ID ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x57
		UCHAR BgmTblId; // ID of the BGM table to set
		UCHAR Mode; // Mode to set the BGM table to
		UCHAR Param1; // Parameter 1 for the BGM table
		UCHAR Param2; // Parameter 2 for the BGM table
		UCHAR Param3; // Parameter 3 for the BGM table
		UCHAR Param4; // Parameter 4 for the BGM table
		} Sce_bgmtbl_set;
		This bytecode sets the specified BGM table parameters.
			08-02-2024 Newly Added

{58} Plc_rot	04	58 ?? ?? 00++
		typedef struct {
		UCHAR Opcode; // 0x58
		UCHAR Rotation; // Rotation value
		UCHAR Speed; // Speed value
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Plc_rot;
		This bytecode sets the specified rotation parameters.
			08-02-2024 Newly Added

{59} Xa_on	04	59 ID ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x59
		UCHAR XaId; // ID of the XA to turn on
		UCHAR Mode; // Mode to set the XA to
		UCHAR Param; // Parameter for the XA
		} Xa_on;
		This bytecode turns on the specified XA with the given parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{5A} Weapon_chg	02	5A ID++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x5A`
		`  UCHAR WeaponId;  // ID of the weapon to change`
		`} Weapon_chg;`
		This bytecode changes the specified weapon parameters.
			08-02-2024 Newly Added

{5B} Plc_cnt	02	5B ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x5B`
		`  UCHAR Count;  // Count value`
		`} Plc_cnt;`
		This bytecode performs the specified arithmetic operation on the counter with the given value.
			08-02-2024 Newly Added

{5C} Sce_shake_on	03	5C ?? ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x5C`
		`  UCHAR Intensity;  // Shake intensity`
		`  UCHAR Duration;  // Shake duration`
		`} Sce_shake_on;`
		This bytecode turns on the specified shake effect with the given parameters.
			08-02-2024 Newly Added

{5D} Mizu_div_set	02	5D ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x5D`
		`  UCHAR Value;  // Value for Mizu_div`
		`} Mizu_div_set;`
		This bytecode sets the specified water division parameters.
			08-02-2024 Newly Added

{5E} Keep_Item_ck	02	5E ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x5E`
		`  UCHAR ItemId;  // ID of the item to check`
		`} Keep_Item_ck;`
		This bytecode checks the specified item parameters.
			08-02-2024 Newly Added

{5F} Xa_vol	02	5F ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x5F`
		`  UCHAR Volume;  // XA volume level`
		`} Xa_vol;`
		This bytecode sets the specified XA volume and pan parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{60} Kage_set	14	60 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x60
		UCHAR Type; // Kage type
		UCHAR Red; // Red color value
		UCHAR Green; // Green color value
		UCHAR Blue; // Blue color value
		UCHAR Alpha; // Alpha transparency value
		UCHAR Flag; // Flag
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		SHORT Width; // Width
		SHORT Height; // Height
		UCHAR zAlign; // Always Zero (Alignment byte)
		} Kage_set;
		This bytecode sets the specified shadow parameters.
			08-02-2024 Newly Added

{61} Cut_be_set	04	61 ?? ?? 00++
		typedef struct {
		UCHAR Opcode; // 0x61
		UCHAR Type; // Cut be type
		USHORT Value; // Value
		} Cut_be_set;
		This bytecode sets the specified cutscene parameters.
			08-02-2024 Newly Added

{62} Sce_Item_lost	02	62 ID++
		typedef struct {
		UCHAR Opcode; // 0x62
		UCHAR ItemId; // ID of the lost item
		} Sce_Item_lost;
		This bytecode removes the specified item from the inventory.
			08-02-2024 Newly Added

{63} Plc_gun_eff	01	63++
		typedef struct {
		UCHAR Opcode; // 0x63
		} Plc_gun_eff;
		This bytecode sets the specified gun effect parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{64} Sce_espr_on2	16	64 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x64
		UCHAR Type; // Type
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		UCHAR Red; // Red color value
		UCHAR Green; // Green color value
		UCHAR Blue; // Blue color value
		UCHAR Alpha; // Alpha transparency value
		} Sce_espr_on2;
		This bytecode turns on the specified espr with the given parameters.
			08-02-2024 Newly Added

{65} Sce_espr_kill2	02	65 ??++
		typedef struct {
		UCHAR Opcode; // 0x65
		UCHAR EsprId; // ID of the effect sprite to kill
		} Sce_espr_kill2;
		This bytecode kills the specified espr with the given parameters.
			08-02-2024 Newly Added

{66} Plc_stop	01	66++
		typedef struct {
		UCHAR Opcode; // 0x66
		} Plc_stop;
		This bytecode stops the specified plc function.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{67} Aot_set_4p	28	67 ??++
		typedef struct {
		UCHAR Opcode; // 0x67
		UCHAR Aot; // AOT index
		UCHAR Id; // ID
		UCHAR Type; // Type
		UCHAR Floor; // Floor
		UCHAR Super; // Super
		SHORT X; // X coordinate
		SHORT Z; // Z coordinate
		SHORT Width; // Width
		SHORT Depth; // Depth
		SHORT Unknown[8]; // Unknown data
		} Aot_set_4p;
		This bytecode sets the specified 4-point AOT parameters.
			08-02-2024 Newly Added

{68} Door_aot_set_4p	40	68 ??++
		typedef struct {
		UCHAR Opcode; // 0x68
		UCHAR Aot; // AOT index
		UCHAR Id; // ID
		UCHAR Type; // Type
		UCHAR Floor; // Floor
		UCHAR Super; // Super
		SHORT X; // X coordinate
		SHORT Z; // Z coordinate
		SHORT Width; // Width
		SHORT Depth; // Depth
		UCHAR Destination; // Destination
		UCHAR NextFloor; // Next floor
		UCHAR NextRoom; // Next room
		UCHAR NextX; // Next X coordinate
		UCHAR NextZ; // Next Z coordinate
		UCHAR NextWidth; // Next width
		UCHAR NextDepth; // Next depth
		SHORT Unknown[12]; // Unknown data
		} Door_aot_set_4p;
		This bytecode sets the specified 4-point door AOT parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{69} Item_aot_set_4p	30	69 ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x69`
		`  UCHAR Aot;  // AOT index`
		`  UCHAR Id;  // ID`
		`  UCHAR Type;  // Type`
		`  UCHAR Floor;  // Floor`
		`  UCHAR Super;  // Super`
		`  SHORT X;  // X coordinate`
		`  SHORT Z;  // Z coordinate`
		`  SHORT Width;  // Width`
		`  SHORT Depth;  // Depth`
		`  SHORT ItemId;  // Item ID`
		`  SHORT Unknown[8];  // Unknown data`
		`} Item_aot_set_4p;`
		This bytecode sets the specified 4-point item AOT parameters.
			08-02-2024 Newly Added

{6A} Light_pos_set	06	6A ID XX XX ZZ ZZ++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x6A`
		`  UCHAR LightId;  // ID of the light source`
		`  SHORT PosX;  // X position`
		`  SHORT PosZ;  // Z position`
		`} Light_pos_set;`
		This bytecode sets the specified light position parameters.
			08-02-2024 Newly Added

{6B} Light_kido_set	04	6B ?? ?? ??++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x6B`
		`  UCHAR LightId;  // ID of the light source`
		`  UCHAR Brightness;  // Brightness value`
		`  UCHAR Unknown;  // Unknown data`
		`} Light_kido_set;`
		This bytecode sets the specified light intensity and color parameters.
			08-02-2024 Newly Added

{6C} Rbj_reset	01	6C++
		`typedef struct {`
		`  UCHAR Opcode;  // 0x6C`
		`} Rbj_reset;`
		This bytecode resets the specified rbj parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{6D} Sce_scr_move	04	6D ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x6D
		UCHAR ScreenId; // Screen ID
		UCHAR Direction; // Direction of movement
		UCHAR Speed; // Speed of movement
		} Sce_scr_move;
		This bytecode moves the specified screen with the given parameters.
			08-02-2024 Newly Added

{6E} Parts_set	06	6E ?? ?? XX XX ZZ ZZ++
		typedef struct {
		UCHAR Opcode; // 0x6E
		UCHAR PartId; // ID of the part
		SHORT PosX; // X position
		SHORT PosZ; // Z position
		} Parts_set;
		This bytecode sets the specified parts parameters.
			08-02-2024 Newly Added

{6F} Movie_on	02	6F ??++
		typedef struct {
		UCHAR Opcode; // 0x6F
		UCHAR MovieId; // ID of the movie
		} Movie_on;
		This bytecode plays the specified movie.
			08-02-2024 Newly Added

{70} Splc_ret	01	70++
		typedef struct {
		UCHAR Opcode; // 0x70
		} Splc_ret;
		This bytecode returns from the specified splc function.
			08-02-2024 Newly Added

{71} Splc_sce	01	71++
		typedef struct {
		UCHAR Opcode; // 0x71
		} Splc_sce;
		This bytecode sets the specified splc parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{72} Super_on	16	72 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x72
		UCHAR Type; // Super type
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		UCHAR Color; // Color
		UCHAR Alpha; // Alpha transparency
		UCHAR Unknown; // Unknown data
		} Super_on;
		This bytecode turns on the specified super with the given parameters.
			08-02-2024 Newly Added

{73} Mirror_set	08	73 ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x73
		UCHAR MirrorId; // ID of the mirror
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		} Mirror_set;
		This bytecode sets the specified mirror position parameters.
			08-02-2024 Newly Added

{74} Sce_fade_adjust	04	74 ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x74
		UCHAR Type; // Fade type
		UCHAR Speed; // Fade speed
		UCHAR Color; // Fade color
		} Sce_fade_adjust;
		This bytecode adjusts the specified fade parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{75} Sce_espr3d_on2	22	75 ?? XX XX YY YY ZZ ZZ DX DX DY DY DZ DZ ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x75
		UCHAR Type; // Type
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		SHORT Scale; // Scale
		SHORT Alpha; // Alpha transparency
		UCHAR R; // Red component
		UCHAR G; // Green component
		UCHAR B; // Blue component
		UCHAR Unknown[6]; // Unknown data
		} Sce_espr3d_on2;
		This bytecode turns on the specified 3D espr with the given parameters.
			08-02-2024 Newly Added

{76} Sce_Item_get	03	76 ?? ?? ID++
		typedef struct {
		UCHAR Opcode; // 0x76
		UCHAR ItemType; // Item type
		UCHAR ItemId; // Item ID
		} Sce_Item_get;
		This bytecode gets the specified item with the given quantity.
			08-02-2024 Newly Added

{77} Sce_line_start	04	77 ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x77
		UCHAR LineId; // Line ID
		UCHAR StartX; // Start X coordinate
		UCHAR StartY; // Start Y coordinate
		} Sce_line_start;
		This bytecode starts the specified line with the given parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{78} Sce_line_main	06	78 ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x78
		UCHAR LineId; // Line ID
		UCHAR StartX; // Start X coordinate
		UCHAR StartY; // Start Y coordinate
		UCHAR EndX; // End X coordinate
		UCHAR EndY; // End Y coordinate
		} Sce_line_main;
		This bytecode sets the specified line parameters.
			08-02-2024 Newly Added

{79} Sce_line_end	01	79++
		typedef struct {
		UCHAR Opcode; // 0x79
		} Sce_line_end;
		This bytecode ends the specified line.
			08-02-2024 Newly Added

{7A} Sce_parts_bomb	16	7A ?? XX XX YY YY ZZ ZZ DX DX DY DY DZ DZ ??++
		typedef struct {
		UCHAR Opcode; // 0x7A
		UCHAR PartId; // Part ID
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		UCHAR Unknown[2]; // Unknown data
		} Sce_parts_bomb;
		This bytecode bombs the specified parts with the given parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{7B} Sce_parts_down	16	7B ?? XX XX YY YY ZZ ZZ DX DX DY DY DZ DZ ??++
		typedef struct {
		UCHAR Opcode; // 0x7B
		UCHAR PartId; // Part ID
		SHORT PosX; // X position
		SHORT PosY; // Y position
		SHORT PosZ; // Z position
		SHORT DirX; // X direction
		SHORT DirY; // Y direction
		SHORT DirZ; // Z direction
		UCHAR Unknown[2]; // Unknown data
		} Sce_parts_down;
		This bytecode moves down the specified parts with the given parameters.
			08-02-2024 Newly Added

{7C} Light_color_set	06	7C 00 XX XX YY YY++
		typedef struct {
		UCHAR Opcode; // 0x7C
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT ColorR; // Red color component of the light
		SHORT ColorG; // Green color component of the light
		SHORT ColorB; // Blue color component of the light
		} Light_color_set;
		This bytecode sets the specified light color parameters.
			08-02-2024 Newly Added

{7D} Light_pos_set2	06	7D ?? XX XX ZZ ZZ++
		typedef struct {
		UCHAR Opcode; // 0x7D
		UCHAR LightId; // ID of the light source
		SHORT PosX; // X position
		SHORT PosZ; // Z position
		} Light_pos_set2;
		This bytecode sets the specified light position parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{7E} Light_kido_set2	06	7E ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x7E
		UCHAR LightId; // ID of the light source
		UCHAR Brightness; // Brightness value
		UCHAR Unknown[4]; // Unknown data
		} Light_kido_set2;
		This bytecode sets the specified light intensity and color parameters.
			08-02-2024 Newly Added

{7F} Light_color_set2	06	7F 00 XX XX YY YY++
		typedef struct {
		UCHAR Opcode; // 0x7F
		UCHAR zAlign; // Always Zero (Alignment byte)
		SHORT ColorR; // Red color component of the light
		SHORT ColorG; // Green color component of the light
		SHORT ColorB; // Blue color component of the light
		} Light_color_set2;
		This bytecode sets the specified light color parameters.
			08-02-2024 Newly Added

{80} Se_vol	02	80 ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x80
		UCHAR Volume; // Volume level
		} Se_vol;
		This bytecode sets the specified sound effect volume and pan parameters.
			08-02-2024 Newly Added

{81} Keep_Item_ck2	03	81 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x81
		UCHAR ItemId; // ID of the item to check
		UCHAR Quantity; // Quantity of the item to check
		} Keep_Item_ck2;
		This bytecode checks the specified item parameters.
			08-02-2024 Newly Added

{82} Sce_espr_task	03	82 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x82
		UCHAR TaskId; // ID of the espr task to set
		UCHAR Mode; // Mode to set the espr task to
		} Sce_espr_task;
		This bytecode sets the specified espr task parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{83} Plc_heal	01	83++
		typedef struct {
		UCHAR Opcode; // 0x83
		} Plc_heal;
		This bytecode heals the specified plc function.
			08-02-2024 Newly Added

{84} St_map_hint	02	84 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x84
		UCHAR HintId; // Hint ID
		} St_map_hint;
		This bytecode sets the specified map hint parameters.
			08-02-2024 Newly Added

{85} Sce_em_pos_ck	06	85 ID XX XX YY YY++
		typedef struct {
		UCHAR Opcode; // 0x85
		UCHAR EntityId; // ID of the entity
		SHORT PosX; // X position
		SHORT PosY; // Y position
		} Sce_em_pos_ck;
		This bytecode checks the specified enemy position parameters.
			08-02-2024 Newly Added

{86} Poison_ck	01	86++
		typedef struct {
		UCHAR Opcode; // 0x86
		} Poison_ck;
		This bytecode checks the specified poison parameters.
			08-02-2024 Newly Added

{87} Poison_clr	01	87++
		typedef struct {
		UCHAR Opcode; // 0x87
		} Poison_clr;
		This bytecode clears the specified poison parameters.
			08-02-2024 Newly Added

{88} Sce_Item_lost2	03	88 ID ??++
		typedef struct {
		UCHAR Opcode; // 0x88
		UCHAR ItemId; // ID of the item
		UCHAR Quantity; // Quantity of the item
		} Sce_Item_lost2;
		This bytecode removes the specified item from the inventory.
			08-02-2024 Newly Added

{89} Evt_next2	01	89++
		typedef struct {
		UCHAR Opcode; // 0x89
		} Evt_next2;
		This bytecode moves to the next event in the sequence.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{8A} Vib_set0	06	8A ID VA LU VA LU++
		typedef struct {
		UCHAR Opcode; // 0x8A
		UCHAR Id; // Vibration ID
		USHORT Value1; // Vibration Value 1
		USHORT Value2; // Vibration Value 2
		} Vib_set0;
		This bytecode sets the specified vibration parameters.
			08-02-2024 Newly Added

{8B} Vib_set1	06	8B ID VA LU VA LU++
		typedef struct {
		UCHAR Opcode; // 0x8B
		UCHAR Id; // Vibration ID
		USHORT Value1; // Vibration Value 1
		USHORT Value2; // Vibration Value 2
		} Vib_set1;
		This bytecode sets the specified vibration parameters.
			08-02-2024 Newly Added

{8C} Vib_fade_set	08	8C ID VA LU VA LU CO NT++
		typedef struct {
		UCHAR Opcode; // 0x8C
		UCHAR VibId; // Vibration ID
		USHORT StartValue; // Starting Vibration Value
		USHORT EndValue; // Ending Vibration Value
		USHORT Count; // Duration of the fade
		} Vib_fade_set;
		This bytecode sets the specified vibration fade parameters.
			08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History

{8D} Item_aot_set2	24	8D ID ?? ?? ?? ?? ?? XX XX ZZ ZZ WW WW DD DD ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x8D
		UCHAR Aot; // Id
		UCHAR SCE; // Type (0x02)
		UCHAR SAT; // Atari
		UCHAR nFloor; // Height
		UCHAR Super; // Matrix
		SHORT X; // Position
		SHORT Z; // Position
		USHORT W; // Size
		USHORT D; // Size
		USHORT iItem; // Item Id
		USHORT nItem; // Item Amount
		USHORT Flag; // argv[1] for Flg_on() and Flg_ck()
		UCHAR MD1; // MD1 File Id - 0xFF if no MD1
		UCHAR Action; // Predefined action taken to obtain
		UCHAR data16; //
		UCHAR data17; //
		} Item_aot_set2;
		This bytecode sets the specified item AOT parameters.
			08-02-2024 Newly Added

{8E} Sce_em_set2	24	8E 00 ID ?? ?? SI ZE ?? ?? ?? XX XX YY YY ZZ ZZ DY DY ?? ?? ?? ??++
		typedef struct {
		UCHAR Opcode; // 0x8E
		UCHAR zAlign; // Alignment byte, always zero
		UCHAR Aot; //
		UCHAR EMD; //
		USHORT Type; //
		UCHAR nFloor; //
		UCHAR SeType; //
		UCHAR ModelType; //
		UCHAR EmSetFlg; //
		SHORT X; //
		SHORT Y; //
		SHORT Z; //
		SHORT DirY; //
		USHORT Timer0; //
		USHORT Timer1; //
		USHORT data16; //
		} Sce_em_set2;
		This bytecode sets the specified enemy parameters.
			08-02-2024 Newly Added