

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {00} Nop         | 01     | <p><b>00++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x00<br/>                     } Nop;<br/>                     This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</p>  | 08-02-2024 Newly Added |
| {01} Evt_end     | 02     | <p><b>01 00++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x01<br/>                     UCHAR zAlign; //<br/>                     Always Zero<br/>                     } Evt_end;<br/>                     This bytecode ends the current Main/Sub script.</p>  | 08-02-2024 Newly Added |
| {02} Evt_next    | 01     | <p><b>02++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x02<br/>                     } Evt_next;<br/>                     This bytecode moves to the next event.</p>  | 08-02-2024 Newly Added |
| {03} Evt_chain   | 02     | <p><b>03 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x03<br/>                     UCHAR Data1; //<br/>                     Data<br/>                     } Evt_chain;<br/>                     This bytecode chains the next event.</p>   | 08-02-2024 Newly Added |
| {04} Evt_exec    | 04     | <p><b>04 ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x04<br/>                     UCHAR Data1; //<br/>                     Data<br/>                     UCHAR Data2; //<br/>                     Data<br/>                     UCHAR Data3; //<br/>                     Data<br/>                     } Evt_exec;<br/>                     This bytecode executes the event.</p> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {05} Evt_kill    | 02     | <p><b>05 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x05<br/>                     UCHAR Data1; //<br/>                     Data<br/>                     } Evt_kill;<br/>                     This bytecode kills the event.</p>  | 08-02-2024 Newly Added |
| {06} Ifel_ck     | 04     | <p><b>06 ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x06<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     USHORT data2; //<br/>                     Data<br/>                     } Ifel_ck;<br/>                     This bytecode checks the condition of an If-Else block.</p> | 08-02-2024 Newly Added |
| {07} Else_ck     | 04     | <p><b>07 ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x07<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     USHORT data2; //<br/>                     Data<br/>                     } Else_ck;<br/>                     This bytecode checks the condition of an Else block.</p>    | 08-02-2024 Newly Added |
| {08} Endif       | 02     | <p><b>08 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x08<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     } Endif;<br/>                     This bytecode ends an If-Else block.</p>  | 08-02-2024 Newly Added |
| {09} Sleep       | 04     | <p><b>09 ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x09<br/>                     UCHAR Data1; //<br/>                     Data<br/>                     USHORT data2; //<br/>                     Data<br/>                     } Sleep;<br/>                     This bytecode pauses the event for a specified duration.</p>             | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {0A} Sleeping    | 03     | <p><b>0A ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x0A<br/>                     USHORT data2; //<br/>                     Data<br/>                     } Sleeping;<br/>                     This bytecode sets the sleeping state.</p>   | 08-02-2024 Newly Added |
| {0B} Wsleep      | 01     | <p><b>0B++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x0B<br/>                     } Wsleep;<br/>                     This bytecode sets the wake sleep state.</p>   | 08-02-2024 Newly Added |
| {0C} Wsleeping   | 01     | <p><b>0C++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x0C<br/>                     } Wsleeping;<br/>                     This bytecode sets the wake sleeping state.</p>   | 08-02-2024 Newly Added |
| {0D} For         | 06     | <p><b>0D ?? ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x0D<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     SHORT data2; //<br/>                     Data<br/>                     USHORT data4; //<br/>                     Data<br/>                     } For;<br/>                     This bytecode starts a For loop.</p> | 08-02-2024 Newly Added |
| {0E} Next        | 02     | <p><b>0E ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x0E<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     } Next;<br/>                     This bytecode ends a For loop.</p>  | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {0F} While       | 04     | <p><b>0F ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x0F<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     SHORT data1; //<br/>                     Data<br/>                     } While;<br/>                     This bytecode starts a While loop.</p> | 08-02-2024 Newly Added |
| {10} Ewhile      | 02     | <p><b>10 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x10<br/>                     UCHAR data1; //<br/>                     Data<br/>                     } Ewhile;<br/>                     This bytecode ends a While loop.</p>  | 08-02-2024 Newly Added |
| {11} Do          | 04     | <p><b>11 ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x11<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     SHORT data2; //<br/>                     Data<br/>                     } Do;<br/>                     This bytecode starts a Do loop.</p>       | 08-02-2024 Newly Added |
| {12} Edwhile     | 02     | <p><b>12 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x12<br/>                     UCHAR data1; //<br/>                     Data<br/>                     } Edwhile;<br/>                     This bytecode ends a Do-While loop.</p>  | 08-02-2024 Newly Added |
| {13} Switch      | 04     | <p><b>13 ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x13<br/>                     UCHAR data1; //<br/>                     Data<br/>                     USHORT data2; //<br/>                     Data<br/>                     } Switch;<br/>                     This bytecode starts a Switch statement.</p>    | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {14} Case        | 06     | <pre>14 ?? ?? ?? ?? ?? ++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x14     UCHAR data1;         // Data     USHORT data2;        // Data } Case; This bytecode defines a Case in a Switch statement.</pre>  | 08-02-2024 Newly Added |
| {15} Default     | 02     | <pre>15 ?? ++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x15     UCHAR zAlign;        // Alignment byte } Default; This bytecode defines the Default case in a Switch statement.</pre>  | 08-02-2024 Newly Added |
| {16} Eswitch     | 02     | <pre>16 ?? ++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x16     UCHAR zAlign;        // Alignment byte } Eswitch; This bytecode ends a Switch statement.</pre>   | 08-02-2024 Newly Added |
| {17} Goto        | 06     | <pre>17 ?? ?? ?? ?? ?? ++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x17     UCHAR data1;         // Data     UCHAR data2;         // Data     UCHAR zAlign;        // Alignment byte     SHORT data4;         // Data } Goto; This bytecode performs a Goto operation.</pre> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {18} Gosub       | 02     | <p><b>18 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x18<br/>                         UCHAR data1; //<br/>                     Data<br/>                     } Gosub;<br/>                     This bytecode performs a Gosub operation.</p>  | 08-02-2024 Newly Added |
| {19} Return      | 02     | <p><b>19 ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x19<br/>                         UCHAR data[3]; //<br/>                     Data<br/>                     } Return;<br/>                     This bytecode returns from a subroutine.</p>  | 08-02-2024 Newly Added |
| {1A} Break       | 02     | <p><b>1A ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x1A<br/>                         CHAR data1; //<br/>                     Data<br/>                     } Break;<br/>                     This bytecode breaks out of a loop.</p>   | 08-02-2024 Newly Added |
| {1B} For2        | 06     | <p><b>1B ?? ?? ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x1B<br/>                         UCHAR zAlign0; //<br/>                     Alignment byte<br/>                         SHORT data2; //<br/>                     Data<br/>                         UCHAR zAlign1; //<br/>                     Alignment byte<br/>                         UCHAR data5; //<br/>                     Data<br/>                     } For2;<br/>                     This bytecode starts a secondary For loop.</p> | 08-02-2024 Newly Added |
| {1C} Break_point | 01     | <p><b>1C++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x1C<br/>                     } Break_point;<br/>                     This bytecode sets a breakpoint.</p>   | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {1D} Work_copy   | 04     | <pre> 1D ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x1D     UCHAR Source;        // Source     UCHAR Destination;  // Destination     UCHAR Typecast;     // Typecast } Work_copy; This bytecode copies work data.                     </pre> | 08-02-2024 Newly Added |
| {1E} Nop1E       | 01     | <pre> 1E++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x1E } Nop1E; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.                     </pre>  | 08-02-2024 Newly Added |
| {1F} Nop1F       | 01     | <pre> 1F++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x1F } Nop1F; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.                     </pre>  | 08-02-2024 Newly Added |
| {20} Nop         | 01     | <pre> 20++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x20 } Nop; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.                     </pre>  | 08-02-2024 Newly Added |
| {21} Ck          | 04     | <pre> 21 ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x21     UCHAR Flag;          // System_flg, etc     UCHAR Id;            // Bit     UCHAR OnOff;         // On/Off } Ck; This bytecode checks a flag.                     </pre>          | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {22} Set         | 04     | <p>22 ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x22<br/>                         UCHAR Flag; //<br/>                     System_flg, etc<br/>                         UCHAR Id; //<br/>                     Bit<br/>                         UCHAR OnOff; //<br/>                     On/Off<br/>                     } Set;<br/>                     This bytecode sets a flag.</p>       | 08-02-2024 Newly Added |
| {23} Cmp         | 06     | <p>23 ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x23<br/>                         UCHAR Flag; //<br/>                     Flag<br/>                         UCHAR Operator; //<br/>                     Operator<br/>                         SHORT Value; //<br/>                     Value<br/>                     } Cmp;<br/>                     This bytecode compares values.</p> | 08-02-2024 Newly Added |
| {24} Save        | 04     | <p>24 ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x24<br/>                         UCHAR Destination; //<br/>                     Destination<br/>                         SHORT Source; //<br/>                     Source<br/>                     } Save;<br/>                     This bytecode saves data.</p>  | 08-02-2024 Newly Added |
| {25} Copy        | 03     | <p>25 ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                         UCHAR Opcode; //<br/>                     0x25<br/>                         UCHAR Source; //<br/>                     Source operand<br/>                         UCHAR Destination; //<br/>                     Destination operand<br/>                     } Copy;<br/>                     This bytecode copies a value from source to destination.</p>                              | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {26} Calc        | 06     | <p>26 ?? ?? ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x26<br/>                     UCHAR Operation; //<br/>                     Operation type<br/>                     UCHAR Operand1; //<br/>                     First operand<br/>                     UCHAR Operand2; //<br/>                     Second operand<br/>                     UCHAR Result; //<br/>                     Result operand<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     } Calc;<br/>                     This bytecode performs a calculation.</p> | 08-02-2024 Newly Added |
| {27} Calc2       | 04     | <p>27 ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x27<br/>                     UCHAR Operation; //<br/>                     Operation type<br/>                     UCHAR Operand; //<br/>                     Operand<br/>                     UCHAR Result; //<br/>                     Result operand<br/>                     } Calc2;<br/>                     This bytecode performs a secondary calculation.</p>   | 08-02-2024 Newly Added |
| {28} Sce_rnd     | 01     | <p>28++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x28<br/>                     } Sce_rnd;<br/>                     This bytecode generates a random number.</p>   | 08-02-2024 Newly Added |
| {29} Cut_chg     | 02     | <p>29 ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x29<br/>                     UCHAR CutsceneId; //<br/>                     Cutscene ID<br/>                     } Cut_chg;<br/>                     This bytecode changes the cutscene.</p>  | 08-02-2024 Newly Added |
| {2A} Cut_old     | 01     | <p>2A++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x2A<br/>                     } Cut_old;<br/>                     This bytecode reverts to the old cutscene.</p>   | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {2B} Message_on  | 06     | <pre> 2B ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x2B     UCHAR MessageId; // Message ID     UCHAR PositionX[2]; // X position     UCHAR PositionY[2]; // Y position     UCHAR Duration; // Display duration } Message_on; This bytecode displays a message on the screen.</pre>   | 08-02-2024 Newly Added |
| {2C} Aot_set     | 20     | <pre> 2C ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x2C     UCHAR AotType; // AOT Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR Floor; // Floor     UCHAR Super; // Super     UCHAR X[2]; // X position     UCHAR Z[2]; // Z position     UCHAR W; // Width     UCHAR D; // Depth     UCHAR Flag[2]; // Flags     UCHAR Path[2]; // Path     UCHAR Direction; // Direction     UCHAR Param[6]; // Parameters } Aot_set; This bytecode sets an AOT (Area of Trigger).</pre> | 08-02-2024 Newly Added |

| Instruction Name   | Length | Example / Info  | History                |
|--------------------|--------|---|------------------------|
| {2D} Obj_model_set | 38     | <pre> 2D ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x2D     UCHAR ObjId;           // Object ID     UCHAR ModelId;        // Model ID     UCHAR MotionId;       // Motion ID     UCHAR Unknown1;       // Unknown     UCHAR Unknown2;       // Unknown     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR R[2];            // Rotation     UCHAR Unknown3[26];   // Unknown } Obj_model_set; This bytecode sets an object model.                     </pre> | 08-02-2024 Newly Added |
| {2E} Work_set      | 03     | <pre> 2E ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x2E     UCHAR WorkNo;         // Work number     UCHAR Value;          // Value } Work_set; This bytecode sets a work value.                     </pre>   | 08-02-2024 Newly Added |
| {2F} Speed_set     | 04     | <pre> 2F ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x2F     UCHAR SpeedX;         // Speed X     UCHAR SpeedY;         // Speed Y     UCHAR SpeedZ;         // Speed Z } Speed_set; This bytecode sets the speed of an object.                     </pre>  | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {30} Add_speed   | 01     | <p><b>30++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x30<br/>                     } Add_speed;<br/>                     This bytecode adds to the speed of an object.</p>  | 08-02-2024 Newly Added |
| {31} Add_aspeed  | 01     | <p><b>31++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x31<br/>                     } Add_aspeed;<br/>                     This bytecode adds to the angular speed of an object.</p>   | 08-02-2024 Newly Added |
| {32} Pos_set     | 08     | <p><b>32 ?? ?? ?? ?? ?? ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x32<br/>                     UCHAR X[2]; //<br/>                     X position<br/>                     UCHAR Y[2]; //<br/>                     Y position<br/>                     UCHAR Z[2]; //<br/>                     Z position<br/>                     UCHAR R[2]; //<br/>                     Rotation<br/>                     } Pos_set;<br/>                     This bytecode sets the position of an object.</p>            | 08-02-2024 Newly Added |
| {33} Dir_set     | 08     | <p><b>33 ?? ?? ?? ?? ?? ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x33<br/>                     UCHAR DirX[2]; //<br/>                     Direction X<br/>                     UCHAR DirY[2]; //<br/>                     Direction Y<br/>                     UCHAR DirZ[2]; //<br/>                     Direction Z<br/>                     UCHAR Speed; //<br/>                     Speed<br/>                     } Dir_set;<br/>                     This bytecode sets the direction of an object.</p> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {34} Member_set  | 04     | <pre> 34 ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x34     UCHAR MemberNo;       // Member number     UCHAR Value[2];       // Value     UCHAR zAlign;         // Alignment byte } Member_set; This bytecode sets a member value.                     </pre>   | 08-02-2024 Newly Added |
| {35} Member_set2 | 03     | <pre> 35 ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x35     UCHAR MemberNo;       // Member number     UCHAR Value;          // Value } Member_set2; This bytecode sets a secondary member value.                     </pre>  | 08-02-2024 Newly Added |
| {36} Se_on       | 12     | <pre> 36 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x36     UCHAR SeId;           // Sound effect ID     UCHAR X[2];           // X position     UCHAR Y[2];           // Y position     UCHAR Z[2];           // Z position     UCHAR Volume;         // Volume     UCHAR Pan;            // Pan     UCHAR Pitch;          // Pitch     UCHAR zAlign[2];      // Alignment bytes } Se_on; This bytecode plays a sound effect.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {37} Sca_id_set  | 04     | <pre> 37 ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x37     UCHAR ScaleId;        // Scale ID     UCHAR X[2];           // X scale     UCHAR Y[2];           // Y scale     UCHAR Z[2];           // Z scale } Sca_id_set; This bytecode sets the scale of an object.                     </pre>  | 08-02-2024 Newly Added |
| {38} Flr_set     | 03     | <pre> 38 ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x38     UCHAR FloorNo;        // Floor number     UCHAR Height;         // Height     UCHAR zAlign;         // Alignment byte } Flr_set; This bytecode sets the floor height.                     </pre>   | 08-02-2024 Newly Added |
| {39} Dir_ck      | 08     | <pre> 39 ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x39     UCHAR DirX[2];         // Direction X     UCHAR DirY[2];         // Direction Y     UCHAR DirZ[2];         // Direction Z     UCHAR CheckType;       // Check type     UCHAR zAlign[2];       // Alignment bytes } Dir_ck; This bytecode checks the direction of an object.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {3A} Sce_espr_on | 16     | <pre> 3A ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x3A     UCHAR EspType; // ESPR Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR zAlign[4]; // Alignment bytes } Sce_espr_on; This bytecode activates an ESPR.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name  | Length | Example / Info  | History                |
|-------------------|--------|---|------------------------|
| {3B} Door_aot_set | 32     | <pre> 3B ?? typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x3B     UCHAR DoorId;          // Door ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;           // Width     UCHAR Height;          // Height     UCHAR Depth;           // Depth     UCHAR KeyItem;         // Key item required     UCHAR NextStage[2];    // Next stage     UCHAR Transition;      // Transition effect     UCHAR zAlign[17];      // Alignment bytes } Door_aot_set; This bytecode sets a door AOT (Area of Trigger).                     </pre> | 08-02-2024 Newly Added |
| {3C} Cut_auto     | 02     | <pre> 3C ???+ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x3C     UCHAR CutsceneId;      // Cutscene ID } Cut_auto; This bytecode automatically triggers a cutscene.                     </pre>  | 08-02-2024 Newly Added |
| {3D} Member_copy  | 03     | <pre> 3D ?? ?? ??+ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x3D     UCHAR Source;          // Source member     UCHAR Destination;     // Destination member } Member_copy; This bytecode copies a member value.                     </pre>   | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {3E} Member_cmp  | 06     | <pre> 3E ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x3E     UCHAR MemberNo;       // Member number     UCHAR Operator;       // Comparison operator     UCHAR Value[2];       // Value     UCHAR Result;         // Comparison result } Member_cmp; This bytecode compares a member value.                     </pre>   | 08-02-2024 Newly Added |
| {3F} Plc_motion  | 04     | <pre> 3F ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x3F     UCHAR MotionId;       // Motion ID     UCHAR Speed;          // Speed     UCHAR zAlign[2];      // Alignment bytes } Plc_motion; This bytecode sets the motion of an object.                     </pre>   | 08-02-2024 Newly Added |
| {40} Plc_dest    | 08     | <pre> 40 ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x40     UCHAR X[2];           // X position     UCHAR Y[2];           // Y position     UCHAR Z[2];           // Z position     UCHAR Speed;          // Speed     UCHAR zAlign[2];      // Alignment bytes } Plc_dest; This bytecode sets the destination of an object.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {41} Plc_neck    | 10     | <p>41 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??</p> <p>??++</p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x41     UCHAR NeckId; // Neck ID     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Rotation; // Rotation     UCHAR zAlign[4]; // Alignment bytes } Plc_neck; This bytecode sets the neck parameters of an object.</pre> | 08-02-2024 Newly Added |
| {42} Plc_ret     | 01     | <p>42++</p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x42 } Plc_ret; This bytecode returns the PLC state.</pre>   | 08-02-2024 Newly Added |
| {43} Plc_flg     | 04     | <p>43 ?? ?? ?? ?? ??++</p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x43     UCHAR Flag; // Flag     UCHAR Value; // Value     UCHAR zAlign[2]; // Alignment bytes } Plc_flg; This bytecode sets a PLC flag.</pre>  | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {44} Sce_em_set  | 22     | <pre> 44 ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x44     UCHAR EmType;         // Enemy type     UCHAR Id;             // ID     UCHAR SCE;           // SCE     UCHAR X[2];          // X position     UCHAR Y[2];          // Y position     UCHAR Z[2];          // Z position     UCHAR Width;        // Width     UCHAR Height;       // Height     UCHAR Depth;        // Depth     UCHAR Life;         // Life     UCHAR Speed;        // Speed     UCHAR zAlign[9];    // Alignment bytes } Sce_em_set; This bytecode sets an enemy.                     </pre> | 08-02-2024 Newly Added |
| {45} Col_chg_set | 05     | <pre> 45 ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x45     UCHAR ColorR;         // Red component     UCHAR ColorG;        // Green component     UCHAR ColorB;        // Blue component     UCHAR zAlign;        // Alignment byte } Col_chg_set; This bytecode changes the color settings.                     </pre>  | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {46} Aot_reset   | 10     | <pre> 46 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x46     UCHAR AotType; // AOT Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR Floor; // Floor     UCHAR Super; // Super     UCHAR X[2]; // X position     UCHAR Z[2]; // Z position     UCHAR W; // Width     UCHAR D; // Depth     UCHAR Flag[2]; // Flags     UCHAR Path[2]; // Path     UCHAR Direction; // Direction     UCHAR zAlign[3]; // Alignment bytes } Aot_reset; This bytecode resets an AOT (Area of Trigger).                     </pre> | 08-02-2024 Newly Added |
| {47} Aot_on      | 02     | <pre> 47 ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x47     UCHAR AotId; // AOT ID } Aot_on; This bytecode activates an AOT.                     </pre>   | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {48} Super_set   | 16     | <pre> 48 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x48     UCHAR SuperType;      // Super type     UCHAR Id;              // ID     UCHAR SCE;             // SCE     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;          // Width     UCHAR Height;         // Height     UCHAR Depth;          // Depth     UCHAR zAlign[2];      // Alignment bytes } Super_set; This bytecode sets a super parameter.                     </pre> | 08-02-2024 Newly Added |
| {49} Super_reset | 08     | <pre> 49 ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x49     UCHAR SuperId;        // Super ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR zAlign[2];      // Alignment bytes } Super_reset; This bytecode resets a super parameter.                     </pre>   | 08-02-2024 Newly Added |
| {4A} Plc_gun     | 02     | <pre> 4A ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x4A     UCHAR GunId;          // Gun ID } Plc_gun; This bytecode sets the gun parameters.                     </pre>  | 08-02-2024 Newly Added |

| Instruction Name   | Length | Example / Info  | History                |
|--------------------|--------|---|------------------------|
| {4B} Cut_replace   | 03     | <pre> 4B ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x4B     UCHAR OldCutId; // Old cutscene ID     UCHAR NewCutId; // New cutscene ID     UCHAR zAlign; // Alignment byte } Cut_replace; This bytecode replaces a cutscene.                     </pre> | 08-02-2024 Newly Added |
| {4C} Sce_espr_kill | 05     | <pre> 4C ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x4C     UCHAR EspType; // ESPR Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR zAlign; // Alignment byte } Sce_espr_kill; This bytecode kills an ESPR.                     </pre>      | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info   | History                |
|---------------------|--------|--|------------------------|
| {4D} Door_model_set | 22     | <pre> 4D ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x4D     UCHAR DoorId;          // Door ID     UCHAR ModelId;         // Model ID     UCHAR MotionId;        // Motion ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;           // Width     UCHAR Height;          // Height     UCHAR Depth;           // Depth     UCHAR zAlign[10];      // Alignment bytes } Door_model_set; This bytecode sets a door model.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name  | Length | Example / Info   | History                |
|-------------------|--------|--|------------------------|
| {4E} Item_aot_set | 22     | <pre> 4E ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x4E     UCHAR ItemId; // Item ID     UCHAR AotType; // AOT Type     UCHAR Id; // ID     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR zAlign[10]; // Alignment bytes } Item_aot_set; This bytecode sets an item AOT.                     </pre> | 08-02-2024 Newly Added |
| {4F} Sce_key_ck   | 04     | <pre> 4F ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x4F     UCHAR KeyId; // Key ID     UCHAR Flag; // Flag     UCHAR zAlign; // Alignment byte } Sce_key_ck; This bytecode checks for a key item.                     </pre>   | 08-02-2024 Newly Added |
| {50} Sce_trg_ck   | 04     | <pre> 50 ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x50     UCHAR TriggerId; // Trigger ID     UCHAR Flag; // Flag     UCHAR zAlign; // Alignment byte } Sce_trg_ck; This bytecode checks for a trigger.                     </pre>  | 08-02-2024 Newly Added |

| Instruction Name      | Length | Example / Info  | History                |
|-----------------------|--------|---|------------------------|
| {51} Sce_bgm_control  | 06     | <p>51 ?? ?? ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x51<br/>                     UCHAR BgmId; //<br/>                     BGM ID<br/>                     UCHAR Volume; //<br/>                     Volume<br/>                     UCHAR Pan; //<br/>                     Pan<br/>                     UCHAR Pitch; //<br/>                     Pitch<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     } Sce_bgm_control;<br/>                     This bytecode controls the background music.</p>                     | 08-02-2024 Newly Added |
| {52} Sce_espr_control | 06     | <p>52 ?? ?? ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x52<br/>                     UCHAR EspId; //<br/>                     ESPR ID<br/>                     UCHAR Command; //<br/>                     Command<br/>                     UCHAR Param1; //<br/>                     Parameter 1<br/>                     UCHAR Param2; //<br/>                     Parameter 2<br/>                     UCHAR zAlign; //<br/>                     Alignment byte<br/>                     } Sce_espr_control;<br/>                     This bytecode controls an ESPR.</p>            | 08-02-2024 Newly Added |
| {53} Sce_fade_set     | 06     | <p>53 ?? ?? ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x53<br/>                     UCHAR FadeType; //<br/>                     Fade type<br/>                     UCHAR Duration; //<br/>                     Duration<br/>                     UCHAR ColorR; //<br/>                     Red component<br/>                     UCHAR ColorG; //<br/>                     Green component<br/>                     UCHAR ColorB; //<br/>                     Blue component<br/>                     } Sce_fade_set;<br/>                     This bytecode sets a fade effect.</p> | 08-02-2024 Newly Added |

| Instruction Name   | Length | Example / Info   | History                |
|--------------------|--------|--|------------------------|
| {54} Sce_espr3d_on | 22     | <pre>54 ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x54     UCHAR EspType; // ESPR Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR zAlign[10]; // Alignment bytes } Sce_espr3d_on; This bytecode activates a 3D ESPR.</pre> | 08-02-2024 Newly Added |
| {55} Member_calc   | 06     | <pre>55 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x55     UCHAR MemberNo; // Member number     UCHAR Operation; // Operation type     UCHAR Operand1; // First operand     UCHAR Operand2; // Second operand     UCHAR Result; // Result operand } Member_calc; This bytecode performs a calculation on a member value.</pre>  | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info  | History                |
|---------------------|--------|---|------------------------|
| {56} Member_calc2   | 04     | <p>56 ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x56<br/>                     UCHAR MemberNo; //<br/>                     Member number<br/>                     UCHAR Operation; //<br/>                     Operation type<br/>                     UCHAR Result; //<br/>                     Result operand<br/>                     } Member_calc2;<br/>                     This bytecode performs a secondary calculation on a member value.</p>  | 08-02-2024 Newly Added |
| {57} Sce_bgmtbl_set | 08     | <p>57 ?? ?? ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x57<br/>                     UCHAR BgmId; //<br/>                     BGM ID<br/>                     UCHAR Volume; //<br/>                     Volume<br/>                     UCHAR Pan; //<br/>                     Pan<br/>                     UCHAR Pitch; //<br/>                     Pitch<br/>                     UCHAR zAlign[3]; //<br/>                     Alignment bytes<br/>                     } Sce_bgmtbl_set;<br/>                     This bytecode sets the BGM table.</p> | 08-02-2024 Newly Added |
| {58} Plc_rot        | 04     | <p>58 ?? ?? ?? ??++<br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x58<br/>                     UCHAR RotationX; //<br/>                     Rotation X<br/>                     UCHAR RotationY; //<br/>                     Rotation Y<br/>                     UCHAR RotationZ; //<br/>                     Rotation Z<br/>                     } Plc_rot;<br/>                     This bytecode sets the rotation of an object.</p>  | 08-02-2024 Newly Added |

| Instruction Name  | Length | Example / Info  | History                |
|-------------------|--------|---|------------------------|
| {59} Xa_on        | 04     | <pre>59 ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x59     UCHAR XaId; // XA ID     UCHAR Volume; // Volume     UCHAR Pan; // Pan     UCHAR zAlign; // Alignment byte } Xa_on; This bytecode activates an XA sound.</pre>       | 08-02-2024 Newly Added |
| {5A} Weapon_chg   | 02     | <pre>5A ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x5A     UCHAR WeaponId; // Weapon ID } Weapon_chg; This bytecode changes the weapon.</pre>  | 08-02-2024 Newly Added |
| {5B} Plc_cnt      | 02     | <pre>5B ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x5B     UCHAR Count; // Count } Plc_cnt; This bytecode sets the PLC count.</pre>  | 08-02-2024 Newly Added |
| {5C} Sce_shake_on | 03     | <pre>5C ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x5C     UCHAR ShakeType; // Shake type     UCHAR Duration; // Duration     UCHAR zAlign; // Alignment byte } Sce_shake_on; This bytecode activates a screen shake effect.</pre> | 08-02-2024 Newly Added |

| Instruction Name  | Length | Example / Info   | History                |
|-------------------|--------|--|------------------------|
| {5D} Mizu_div_set | 02     | <p><b>5D ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x5D<br/>                     UCHAR Division; //<br/>                     Division<br/>                     } Mizu_div_set;<br/>                     This bytecode sets the water division.</p>  | 08-02-2024 Newly Added |
| {5E} Keep_Item_ck | 02     | <p><b>5E ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x5E<br/>                     UCHAR ItemId; //<br/>                     Item ID<br/>                     } Keep_Item_ck;<br/>                     This bytecode checks for a kept item.</p>  | 08-02-2024 Newly Added |
| {5F} Xa_vol       | 02     | <p><b>5F ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x5F<br/>                     UCHAR Volume; //<br/>                     Volume<br/>                     } Xa_vol;<br/>                     This bytecode sets the XA volume.</p>   | 08-02-2024 Newly Added |
| {60} Kage_set     | 14     | <p><b>60 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++</b><br/>                     typedef struct { // Ptr //<br/>                     Description<br/>                     UCHAR Opcode; //<br/>                     0x60<br/>                     UCHAR KageType; //<br/>                     Shadow type<br/>                     UCHAR X[2]; //<br/>                     X position<br/>                     UCHAR Y[2]; //<br/>                     Y position<br/>                     UCHAR Z[2]; //<br/>                     Z position<br/>                     UCHAR Width; //<br/>                     Width<br/>                     UCHAR Height; //<br/>                     Height<br/>                     UCHAR Depth; //<br/>                     Depth<br/>                     UCHAR zAlign[2]; //<br/>                     Alignment bytes<br/>                     } Kage_set;<br/>                     This bytecode sets a shadow.</p> | 08-02-2024 Newly Added |

| Instruction Name   | Length | Example / Info  | History                |
|--------------------|--------|---|------------------------|
| {61} Cut_be_set    | 04     | <pre>61 ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x61     UCHAR CutsceneId; // Cutscene ID     UCHAR Flag; // Flag     UCHAR zAlign[2]; // Alignment bytes } Cut_be_set; This bytecode sets a cutscene BE.</pre> | 08-02-2024 Newly Added |
| {62} Sce_Item_lost | 02     | <pre>62 ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x62     UCHAR ItemId; // Item ID } Sce_Item_lost; This bytecode sets an item as lost.</pre>   | 08-02-2024 Newly Added |
| {63} Plc_gun_eff   | 01     | <pre>63++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x63 } Plc_gun_eff; This bytecode sets the gun effect.</pre>  | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info  | History                |
|---------------------|--------|---|------------------------|
| {64} Sce_espr_on2   | 16     | <pre> 64 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x64     UCHAR EspType; // ESPR Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR zAlign[4]; // Alignment bytes } Sce_espr_on2; This bytecode activates a secondary ESPR.                     </pre> | 08-02-2024 Newly Added |
| {65} Sce_espr_kill2 | 02     | <pre> 65 ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x65     UCHAR EspId; // ESPR ID } Sce_espr_kill2; This bytecode kills a secondary ESPR.                     </pre>   | 08-02-2024 Newly Added |
| {66} Plc_stop       | 01     | <pre> 66++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x66 } Plc_stop; This bytecode stops the PLC.                     </pre>   | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {67} Aot_set_4p  | 28     | <pre> 67 ?? typedef struct { // Ptr // Description     UCHAR Opcode; // 0x67     UCHAR AotType; // AOT Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR Floor; // Floor     UCHAR Super; // Super     UCHAR X[2]; // X position     UCHAR Z[2]; // Z position     UCHAR W; // Width     UCHAR D; // Depth     UCHAR Flag[2]; // Flags     UCHAR Path[2]; // Path     UCHAR Direction; // Direction     UCHAR Param[10]; // Parameters } Aot_set_4p; This bytecode sets an AOT (Area of Trigger) with 4 parameters.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name     | Length | Example / Info  | History                |
|----------------------|--------|---|------------------------|
| {68} Door_aot_set_4p | 40     | <pre> 68 ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x68     UCHAR DoorId;          // Door ID     UCHAR AotType;         // AOT Type     UCHAR Id;              // ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;          // Width     UCHAR Height;          // Height     UCHAR Depth;          // Depth     UCHAR KeyItem;         // Key item required     UCHAR NextStage[2];    // Next stage     UCHAR Transition;      // Transition effect     UCHAR Param[12];       // Parameters } Door_aot_set_4p; This bytecode sets a door AOT (Area of Trigger) with 4 parameters.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name     | Length | Example / Info   | History                |
|----------------------|--------|--|------------------------|
| {69} Item_aot_set_4p | 30     | <pre> 69 ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x69     UCHAR ItemId;          // Item ID     UCHAR AotType;         // AOT Type     UCHAR Id;              // ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;          // Width     UCHAR Height;         // Height     UCHAR Depth;          // Depth     UCHAR Param[14];      // Parameters } Item_aot_set_4p; This bytecode sets an item AOT with 4 parameters.                     </pre> | 08-02-2024 Newly Added |
| {6A} Light_pos_set   | 06     | <pre> 6A ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x6A     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position } Light_pos_set; This bytecode sets the position of a light.                     </pre>  | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info  | History                |
|---------------------|--------|---|------------------------|
| {6B} Light_kido_set | 04     | <pre>6B ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x6B     UCHAR Intensity;     // Light intensity     UCHAR ColorR;        // Red component     UCHAR ColorG;        // Green component     UCHAR ColorB;        // Blue component } Light_kido_set; This bytecode sets the light intensity and color.</pre> | 08-02-2024 Newly Added |
| {6C} Rbj_reset      | 01     | <pre>6C++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x6C } Rbj_reset; This bytecode resets the RBJ.</pre>   | 08-02-2024 Newly Added |
| {6D} Sce_scr_move   | 04     | <pre>6D ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x6D     UCHAR ScreenX[2];    // Screen X position     UCHAR ScreenY[2];    // Screen Y position } Sce_scr_move; This bytecode moves the screen position.</pre>   | 08-02-2024 Newly Added |
| {6E} Parts_set      | 06     | <pre>6E ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x6E     UCHAR PartId;        // Part ID     UCHAR X[2];          // X position     UCHAR Y[2];          // Y position     UCHAR Z[2];          // Z position } Parts_set; This bytecode sets the parts.</pre>   | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info  | History                |
|------------------|--------|---|------------------------|
| {6F} Movie_on    | 02     | <p><b>6F ??++</b></p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x6F     UCHAR MovieId; // Movie ID } Movie_on; This bytecode activates a movie.</pre>   | 08-02-2024 Newly Added |
| {70} Splc_ret    | 01     | <p><b>70++</b></p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x70 } Splc_ret; This bytecode returns the SPLC state.</pre>  | 08-02-2024 Newly Added |
| {71} Splc_sce    | 01     | <p><b>71++</b></p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x71 } Splc_sce; This bytecode sets the SPLC SCE.</pre>   | 08-02-2024 Newly Added |
| {72} Super_on    | 16     | <p><b>72 ?? ++</b></p> <pre>typedef struct { // Ptr // Description     UCHAR Opcode; // 0x72     UCHAR SuperType; // Super type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR zAlign[2]; // Alignment bytes } Super_on; This bytecode activates a super parameter.</pre> | 08-02-2024 Newly Added |

| Instruction Name     | Length | Example / Info   | History                |
|----------------------|--------|--|------------------------|
| {73} Mirror_set      | 08     | <pre> 73 ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x73     UCHAR X[2];          // X position     UCHAR Y[2];          // Y position     UCHAR Z[2];          // Z position     UCHAR Reflection;    // Reflection type } Mirror_set; This bytecode sets a mirror.                     </pre>   | 08-02-2024 Newly Added |
| {74} Sce_fade_adjust | 04     | <pre> 74 ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x74     UCHAR FadeType;      // Fade type     UCHAR Duration;      // Duration     UCHAR ColorR;        // Red component     UCHAR ColorG;        // Green component     UCHAR ColorB;        // Blue component } Sce_fade_adjust; This bytecode adjusts the fade effect.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info  | History                |
|---------------------|--------|---|------------------------|
| {75} Sce_espr3d_on2 | 22     | <pre> 75 ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x75     UCHAR EspType; // ESPR Type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR zAlign[10]; // Alignment bytes } Sce_espr3d_on2; This bytecode activates a secondary 3D ESPR.                     </pre> | 08-02-2024 Newly Added |
| {76} Sce_Item_get   | 03     | <pre> 76 ??? ?++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x76     UCHAR ItemId; // Item ID     UCHAR Flag; // Flag     UCHAR zAlign; // Alignment byte } Sce_Item_get; This bytecode gets an item.                     </pre>   | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info   | History                |
|---------------------|--------|--|------------------------|
| {77} Sce_line_start | 04     | <pre> 77 ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x77     UCHAR LineId; // Line ID     UCHAR StartX[2]; // Start X position     UCHAR StartY[2]; // Start Y position     UCHAR StartZ[2]; // Start Z position } Sce_line_start; This bytecode starts a line.                     </pre>  | 08-02-2024 Newly Added |
| {78} Sce_line_main  | 06     | <pre> 78 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x78     UCHAR LineId; // Line ID     UCHAR MainX[2]; // Main X position     UCHAR MainY[2]; // Main Y position     UCHAR MainZ[2]; // Main Z position } Sce_line_main; This bytecode sets the main line.                     </pre> | 08-02-2024 Newly Added |
| {79} Sce_line_end   | 01     | <pre> 79++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x79 } Sce_line_end; This bytecode ends a line.                     </pre>  | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info  | History                |
|---------------------|--------|---|------------------------|
| {7A} Sce_parts_bomb | 16     | <pre> 7A ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x7A     UCHAR PartId;          // Part ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Radius;          // Radius     UCHAR Power;           // Power     UCHAR zAlign[8];       // Alignment bytes } Sce_parts_bomb; This bytecode sets a parts bomb.                     </pre>                                     | 08-02-2024 Newly Added |
| {7B} Sce_parts_down | 16     | <pre> 7B ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x7B     UCHAR PartId;          // Part ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;           // Width     UCHAR Height;          // Height     UCHAR Depth;           // Depth     UCHAR zAlign[6];       // Alignment bytes } Sce_parts_down; This bytecode sets a parts down.                     </pre> | 08-02-2024 Newly Added |

| Instruction Name     | Length | Example / Info  | History                |
|----------------------|--------|---|------------------------|
| {7C} Light_color_set | 06     | <pre> 7C ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x7C     UCHAR ColorR;        // Red component     UCHAR ColorG;        // Green component     UCHAR ColorB;        // Blue component     UCHAR zAlign[3];     // Alignment bytes } Light_color_set; This bytecode sets the color of a light.                     </pre>   | 08-02-2024 Newly Added |
| {7D} Light_pos_set2  | 06     | <pre> 7D ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x7D     UCHAR X[2];          // X position     UCHAR Y[2];          // Y position     UCHAR Z[2];          // Z position } Light_pos_set2; This bytecode sets the position of a light (secondary).                     </pre>   | 08-02-2024 Newly Added |
| {7E} Light_kido_set2 | 06     | <pre> 7E ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x7E     UCHAR Intensity;     // Light intensity     UCHAR ColorR;        // Red component     UCHAR ColorG;        // Green component     UCHAR ColorB;        // Blue component     UCHAR zAlign;        // Alignment byte } Light_kido_set2; This bytecode sets the light intensity and color (secondary).                     </pre> | 08-02-2024 Newly Added |

| Instruction Name      | Length | Example / Info   | History                |
|-----------------------|--------|--|------------------------|
| {7F} Light_color_set2 | 06     | <pre>7F ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x7F     UCHAR ColorR;        // Red component     UCHAR ColorG;        // Green component     UCHAR ColorB;        // Blue component     UCHAR zAlign[3];     // Alignment bytes } Light_color_set2; This bytecode sets the color of a light (secondary).</pre> | 08-02-2024 Newly Added |
| {80} Se_vol           | 02     | <pre>80 ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x80     UCHAR Volume;        // Volume } Se_vol; This bytecode sets the volume of a sound effect.</pre>  | 08-02-2024 Newly Added |
| {81} Sce_Item_cmp     | 03     | <pre>81 ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x81     UCHAR ItemId;        // Item ID     UCHAR Flag;          // Flag     UCHAR zAlign;        // Alignment byte } Sce_Item_cmp; This bytecode compares an item.</pre>  | 08-02-2024 Newly Added |
| {82} Sce_espr_task    | 03     | <pre>82 ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x82     UCHAR EspId;         // ESPR ID     UCHAR TaskId;        // Task ID     UCHAR zAlign;        // Alignment byte } Sce_espr_task; This bytecode assigns a task to an ESPR.</pre>   | 08-02-2024 Newly Added |

| Instruction Name   | Length | Example / Info   | History                |
|--------------------|--------|--|------------------------|
| {83} Plc_heal      | 01     | <b>83++</b><br><pre>typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x83 } Plc_heal; This bytecode heals a PLC.</pre>  | 08-02-2024 Newly Added |
| {84} St_map_hint   | 02     | <b>84 ??++</b><br><pre>typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x84     UCHAR HintId;          // Hint ID } St_map_hint; This bytecode sets a map hint.</pre>  | 08-02-2024 Newly Added |
| {85} Sce_em_pos_ck | 06     | <b>85 ?? ?? ?? ?? ?? ??++</b><br><pre>typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x85     UCHAR EmType;          // Enemy type     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position } Sce_em_pos_ck; This bytecode checks the position of an enemy.</pre> | 08-02-2024 Newly Added |
| {86} Poison_ck     | 01     | <b>86++</b><br><pre>typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x86 } Poison_ck; This bytecode checks for poison status.</pre>  | 08-02-2024 Newly Added |
| {87} Poison_clr    | 01     | <b>87++</b><br><pre>typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x87 } Poison_clr; This bytecode clears the poison status.</pre>   | 08-02-2024 Newly Added |

| Instruction Name    | Length | Example / Info  | History                |
|---------------------|--------|---|------------------------|
| {88} Sce_Item_lost2 | 03     | <pre>88 ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x88     UCHAR ItemId;        // Item ID     UCHAR Flag;          // Flag     UCHAR zAlign;        // Alignment byte } Sce_Item_lost2; This bytecode sets an item as lost (secondary).</pre>   | 08-02-2024 Newly Added |
| {89} Evt_next2      | 01     | <pre>89++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x89 } Evt_next2; This bytecode moves to the next event (secondary).</pre>  | 08-02-2024 Newly Added |
| {8A} Vib_set0       | 06     | <pre>8A ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;        // 0x8A     UCHAR VibrationId;   // Vibration ID     UCHAR Intensity;     // Intensity     UCHAR Duration;      // Duration     UCHAR Frequency;     // Frequency     UCHAR zAlign[2];     // Alignment bytes } Vib_set0; This bytecode sets vibration parameters.</pre> | 08-02-2024 Newly Added |

| Instruction Name  | Length | Example / Info   | History                |
|-------------------|--------|--|------------------------|
| {8B} Vib_set1     | 06     | <pre> 8B ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x8B     UCHAR VibrationId;    // Vibration ID     UCHAR Intensity;      // Intensity     UCHAR Duration;       // Duration     UCHAR Frequency;      // Frequency     UCHAR zAlign[2];      // Alignment bytes } Vib_set1; This bytecode sets secondary vibration parameters.</pre>  | 08-02-2024 Newly Added |
| {8C} Vib_fade_set | 08     | <pre> 8C ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x8C     UCHAR FadeId;         // Fade ID     UCHAR StartIntensity; // Start intensity     UCHAR EndIntensity;   // End intensity     UCHAR Duration;       // Duration     UCHAR Frequency;      // Frequency     UCHAR zAlign[2];      // Alignment bytes } Vib_fade_set; This bytecode sets the fade effect for vibration.</pre> | 08-02-2024 Newly Added |

| Instruction Name   | Length | Example / Info   | History                |
|--------------------|--------|--|------------------------|
| {8D} Item_aot_set2 | 24     | <pre> 8D ??++ typedef struct {          // Ptr // Description     UCHAR Opcode;          // 0x8D     UCHAR ItemId;          // Item ID     UCHAR AotType;         // AOT Type     UCHAR Id;              // ID     UCHAR X[2];            // X position     UCHAR Y[2];            // Y position     UCHAR Z[2];            // Z position     UCHAR Width;          // Width     UCHAR Height;         // Height     UCHAR Depth;          // Depth     UCHAR Param[12];      // Parameters } Item_aot_set2; This bytecode sets an item AOT (secondary).                     </pre> | 08-02-2024 Newly Added |

| Instruction Name | Length | Example / Info   | History                |
|------------------|--------|--|------------------------|
| {8E} Sce_em_set2 | 24     | <pre> 8E ??++ typedef struct { // Ptr // Description     UCHAR Opcode; // 0x8E     UCHAR EmType; // Enemy type     UCHAR Id; // ID     UCHAR SCE; // SCE     UCHAR X[2]; // X position     UCHAR Y[2]; // Y position     UCHAR Z[2]; // Z position     UCHAR Width; // Width     UCHAR Height; // Height     UCHAR Depth; // Depth     UCHAR Life; // Life     UCHAR Speed; // Speed     UCHAR zAlign[9]; // Alignment bytes } Sce_em_set2; This bytecode sets a secondary enemy. </pre> | 08-02-2024 Newly Added |

From: <https://classicmodification.com/> - **Classic RE Modification**

Permanent link: [https://classicmodification.com/doku.php?id=re2\\_opcodes&rev=1722656560](https://classicmodification.com/doku.php?id=re2_opcodes&rev=1722656560)

Last update: **2024/08/02 20:42**

