Event Structure Overview:

For constructing your event.json you must always start out with a “{}” and follow the formatting laid out below within those brackets:

"Events": [

{"OpeningEvent": [ (*Your event commands here*)]},

{“Your other event name”: [ (*Your event commands here*)]},

Etc.

{“EndingEvent”: [(*Your event commands here*), {“ENDA”: true}]}

],

“AreaConditions”: [ {*Your Area Conditions Here*}]

For context, you must always include an “OpeningEvent” and “EndingEvent” within the “Events” scope. And you must always include the Conditional events after the “Events” scope.

All events **MUST ALWAYS** end with a {“ENDA”: true} statement which proceeds after your other commands, otherwise the event will never end!

**Basic Overview of Command Calls for Normal Events:**

LOAD1 – Loads a unit or unit block onto the map.

STAL – pauses the event for a certain amount of time in frames

FADI – fades the screen in black

FADU – fades the screen out from black

TEXTSHOW – Displays a textbox with a text of choice

AUTPARTY – Adds a unit to the party

ADDITEM – Adds a given number of items to the party’s inventory

REMOVE\_ITEM – Removes a selected number of items from the party’s inventory

GIVE\_MONEY – Adds money to your party’s inventory.

REMOVE\_MONEY – Subtracts money from your party’s inventory.

SHOP – Calls the shop to whatever shop you specify. Make sure that it exists in the shop folder in data.

MUSC – Changes the music with a standard fade out effect

MUSFI – Changes the music with a fade out parameter.

MUSNF – Changes the music with no fade.

CHG\_BGM – Change the level’s default background music (Does not change the music currently playing)

RETURN\_TO\_BGM – Changes the music to the level’s default background music.

CHANGE\_LEVEL – Changes the level, along with the map, based on the Level ID in the level database.

TILECHANGE – Activates a tiled layer to be visible.

TILEREVERSE – De-activates a tiled layer back to be invisible.

ADD\_BLOCKER\_SET – Activates a blocker set to allow collision with units on the map when they are on the respective layer.

REMOVE\_BLOCKER\_SET – Removes the blocker set from activating collision with units on the map.

BATTLE – Creates a custom battle event with custom parameters.

CALL – Calls another event within the same event script

CAM1 – Moves the camera either on a unit or to a set location defined within the parameters

CAM2 – Much like CAM1, moves the camera either on a unit or to set location, however it warps automatically to the set location of either the unit or location. Good for using in between map transitions.

MOVE – Moves a desired unit to a select location

REPOS – Repositions the given unit to a set location with the user given coordinates.

CLEAR – Clears all units from the level;

REMOVE\_PARTY – Removes the party from the level;

CLEE – Removes all **enemy** type units from the level;

CLEN – Removes all **ally** type units from the level;

CLEAR\_UNIT – Removes a **specific** unit or units from the level;

REMU – Hides a **specific** unit from the party.

REVEAL – Reveals a unit that is hidden from the party.

DEPART – Allows a **specific** unit from the party to be added to the party reserve.

RETURN\_TO\_PARTY – Allows a **specific** unit from the party reserve to join the party.

CHG\_LAYER – Changes what layer a unit with be drawn on based on the order in the tiled map. Starts at 0 instead of 1.

CHG\_ANIMATION – Changes a specific unit’s map animation. WARNING: This command locks the unit’s map animations to only play what you specify and therefore persists after the event is done playing. Call UNLOCK\_ANIM to make the unit’s animations return to normal.

CHG\_DIRECTION – Changes the direction of the unit on the map.

UNLOCK\_ANIM – Unlocks the map animations to play back the default animations.

SHOW\_VFX – Show some visual effects on the screen, that the user specifies.

PRINT – Prints a string to the Command Console (mostly used for debugging purposes)

LABEL – Declares a point in the event where you can go-to later or branch off to.

BEQ – Compares two flags, two integer values, or one flag to an integer value to see if they equal to each other and will jump to the Label provided in the parameters

ENUT – Sets a flag to true; can be used for local or global flags.

ENUF – Sets a flag to false; can be used for local or global flags.

GOTO – Jumps to a Label declared within the events.

ENDA – Ends the current event and returns to the game.

**Normal Event Parameter Implementation:**

STAL – {“STAL”: (*Integer Value*)}

FADI – {“FADI”: {*Integer Value*}}

FADU – {“FADU”: {*Integer Value*}}

TEXTSHOW – {“TEXTSHOW”: (***Text ID****; type String*)}

AUTPARTY – {“AUTPARTY”: (***Unit’s ID****; type Int*)}

ADDITEM – {“ADDITEM”: {“item”: (***Item file name****; type String*), “amount”: (*Integer Value*)}}

REMOVE\_ITEM – {“REMOVE\_ITEM”: {“item”: (***Item file name****; type String*), “amount”: (*Positive Integer Value*)}}

GIVE\_MONEY – {"GIVE\_MONEY": Int value}

REMOVE\_MONEY – {"REMOVE\_MONEY": Int value}

SHOP – {"SHOP": "Shop Name"}

MUSC – {“MUSC”: (***Music’s Name****; type String*)}; e.g. {MUSC: (“opening\_song”)}

MUSC – {“MUSC”: “null”}; turns the music off.

MUSFI – {“MUSFI”: {“music”: *(****Music’s Name****; type String*), “fade\_out”: (Float Value)}}

MUSNF – {“MUSNF”: (***Music’s Name****; type String*)}}

CHG\_BGM – {“CHG\_BGM”: (***Music’s Name****; type String*)}

RETURN\_TO\_BGM – {“RETURN\_TO\_BGM”: (*Bool Value*)}; e.g. {RETURN\_TO\_BGM: true}

CHANGE\_LEVEL – {"CHANGE\_LEVEL": Level ID value}

TILECHANGE – {“TILECHANGE”: (***Tile Change’s Name;*** *type String*)}

TILEREVERSE – {“TILEREVERSE”: (***Tile Change’s Name;*** *type String*)}

ADD\_BLOCKER\_SET – {“ADD\_BLOCKER\_SET”: (***Blocker Set’s Name***)}

REMOVE\_BLOCKER\_SET – {“REMOVE\_BLOCKER\_SET”: (***Blocker Set’s Name***)}

BATTLE – {“BATTLE”: {“enemy\_party”: (***Enemy party’s name***; *type String*), “music”: (***Music’s name****; type String)*}}

CALL – {“CALL”: (***Event’s name****; type String*)}

CAM1 – {“CAM1”: {“Unit”: (***Unit’s ID****; type Int*), “speed”: (*Int Value*)}}

CAM1 – {"CAM1": {"x": (*Integer Value*), "y": (*Integer Value*), "speed": (*Integer Value*)}}; *speed is an optional parameter.*

CAM2 – {"CAM2": {"Unit": Unit’s ID}} // *Substitute Unit ID for* "party" *to go to the party*

CAM2 – {"CAM2": {"x": Int value, "y": Int value}}

MOVE – {"MOVE": {"Unit": (***Unit’s name****; type String*) "speed": (*Integer Value*), "x": (*Integer Value*), "y": (*Integer Value*)}}

REPOS – {"REPOS": {"Unit": Unit’s ID, "Location": {"x": Int value, "y": Int value}, "cam2": bool value}} // *Substitute Unit ID for* "party" *to modify the party’s location. The “cam2” parameter is optional when including this command.*

CLEAR – {"CLEAR": bool value} // *must be set to* true *for it work*

REMOVE\_PARTY – {"REMOVE\_PARTY": bool value} // *must be set to* true *for it work*

CLEE – {"CLEE": bool value} // *must be set to* true *for it work*

CLEN – {"CLEN": bool value} // *must be set to* true *for it work*

CLEAR\_UNIT – {"CLEAR\_UNIT": Unit ID, "ClearAll": bool value} // *The* "ClearAll" *parameter is optional.*

CLEAR\_UNIT – {"CLEAR\_UNIT": Unit position in list, "ByPosition": bool value} // *The* "ByPosition" ***is required*** *and must be set to* true *for clearing by position to work.*

REMU – {"REMU": Unit ID}

REVEAL – {"REVEAL": Unit ID}

DEPART – {"DEPART": Unit ID, "Clear": false} // *The* "Clear" *parameter is optional.*

RETURN\_TO\_PARTY – {"RETURN\_TO\_PARTY": Unit ID}

CHG\_LAYER – {“CHG\_LAYER”: {“Unit”: (***Unit’s ID****; type Int*), “Layer”: (***Layer pos number;*** *type Int*)}}; *Type “party” in the Unit parameter to change the party’s layer position.*

CHG\_ANIM – {“CHG\_ANIM”: {“Unit”: (***Unit’s ID****; type Int*), “Animation”: (***Animation’s name;*** *type String*), “Flip”: (***true or false***)}}; *Type “party” in the Unit parameter to change the party’s animations.*

CHG\_DIRECTION – {“CHG\_DIRECTION”: {“Unit”: (***Unit’s ID****; type Int*), “Direction”: (***Direction****; type String*)}}; *Type “party” in the Unit parameter to change the party’s map direction. Direction types: up, down, left, and right.*

UNLOCK\_ANIM – {“UNLOCK\_ANIM”: {“Unit”: (***Unit’s name;*** *type String*)}}; *Type “party” in the Unit parameter to unlock the party’s map animations.*

SHOW\_VFX – {"SHOW\_VFX": "vfx animation name"}

PRINT – {“PRINT”: (*String Value*)}

LABEL – {“LABEL”: (*Integer Value*)}

BEQ – {“BEQ”: {“VALUE1”: *(Integer Value or Flag*), “VALUE2”: *(Integer Value or Flag*), “GOTO”: (***Label ID****; Type Integer*)}}; *Flag is always of type String*

ENUT – {"ENUT": "*Flag Name*"} *or* {"ENUT": "*Flag Name*", "IsGlobal": true}

ENUF – {"ENUF": "*Flag Name*"} *or* {"ENUF": "*Flag Name*", "IsGlobal": true}

GOTO – {“GOTO”: (*Integer Value*)}

ENDA – {“ENDA”: true}