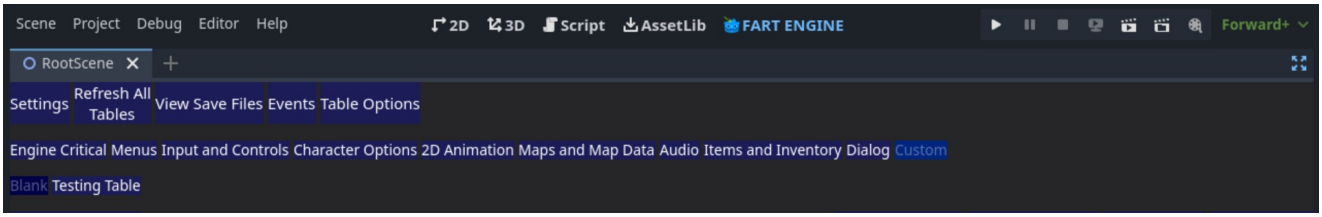
Fart Engine Documentation

Main Screen Navigation



Top Row

Contains engine top level options

Refresh All Tables: Reloads the engine display and data without savings any changes

View Save Files: Shows any game save files and allows for editing of saved data

Events: Contains all of the tools for creating events or anything that the player can interact with

Table Options: Contains a list of all the tables in the engine as well as several customization options

Settings: Contains various settings for current project such as the save locations for tables and which game profile to use

Middle Row

Contains table category names. Select category button to populate all table buttons associated with selected category

Bottom Row

List of tables assigned to the category selected in Middle Row. Selecting a table will display the selected table in the Table Display

Table Display

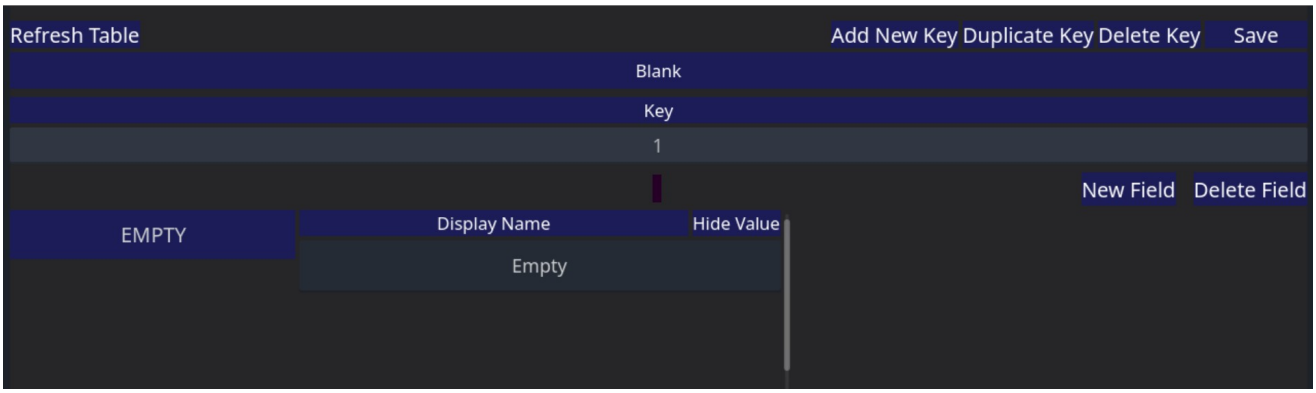


Table Display UI Elements

Refresh Table: Reloads the data for the current table. This will reset any field values that have been modified but not saved

Add New Key: Adds a new key to the selected table. Will use the next number as the primary key but will prompt the user for a Display Name. All fields and their values will be copied from the first key of the table

Duplicate Key: This will add a new key but user is not prompted for Display Name. The key number is used as the Display Name and all fields and their values are copied from selected key to duplicate new key

Delete Key: Removes the selected Key from current table. User will be prompted to confirm deletion

Save: Saves the data for the current table and updates the engine

Key Selection Buttons: When selected, the engine will update the Table Display Area with the fields and values associated with the selected Key

Show/Hide Value: If true, the field will show the node's input area, if false the Field Name Label will still be visible but the current value will be hidden. Linked to Show Value in selected field options

Data Change Notification: Will display a warning message when field values have been changed but the table has not yet been saved

Display Name: A special field that is in all tables and should not be deleted. If table option Is Set To Use Display Name in the editor is true for selected table, the engine will use this value instead of the selected primary key when referencing selected Key

Rearranging Display Order of Table Keys

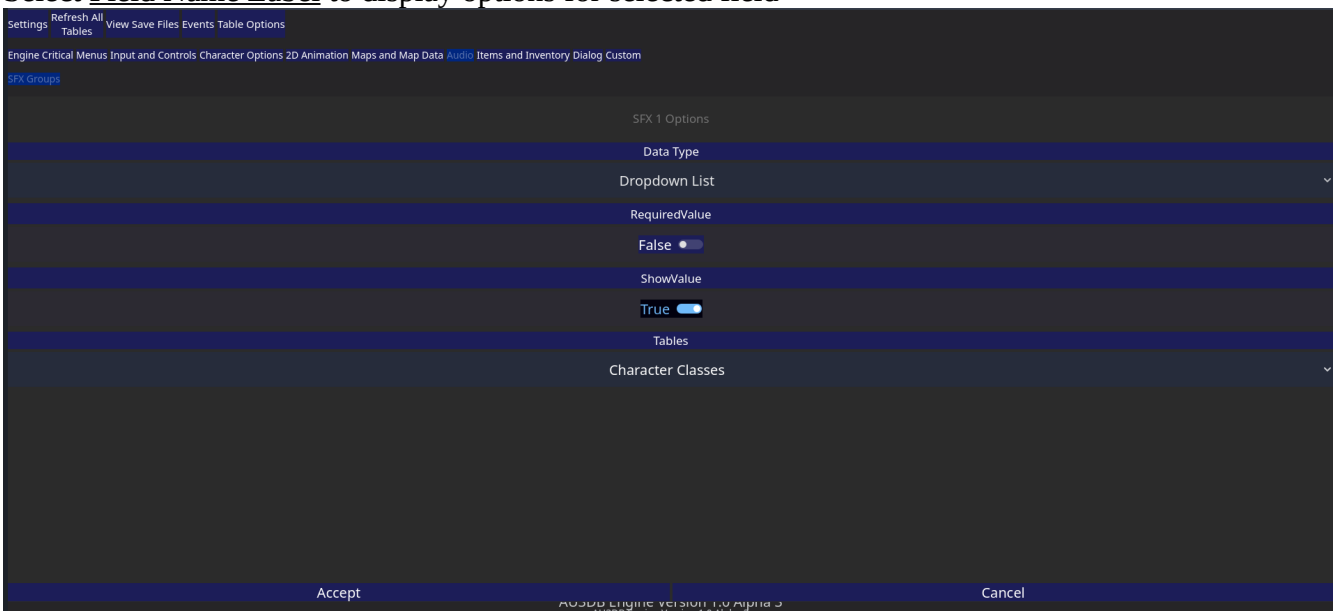
Hold desired Key Selection Button down until it detaches from the list. The Key will be inserted below the next Key Selection Button that is selected

Table Settings

To change individual table settings, see Table Options section

Field Name Label

Select Field Name Label to display options for selected field



Data Type: Allows user to modify the Data Type of selected field. Changing this will reset the selected field's value to default for new Data Type

Required Value: If true, the selected field cannot be deleted

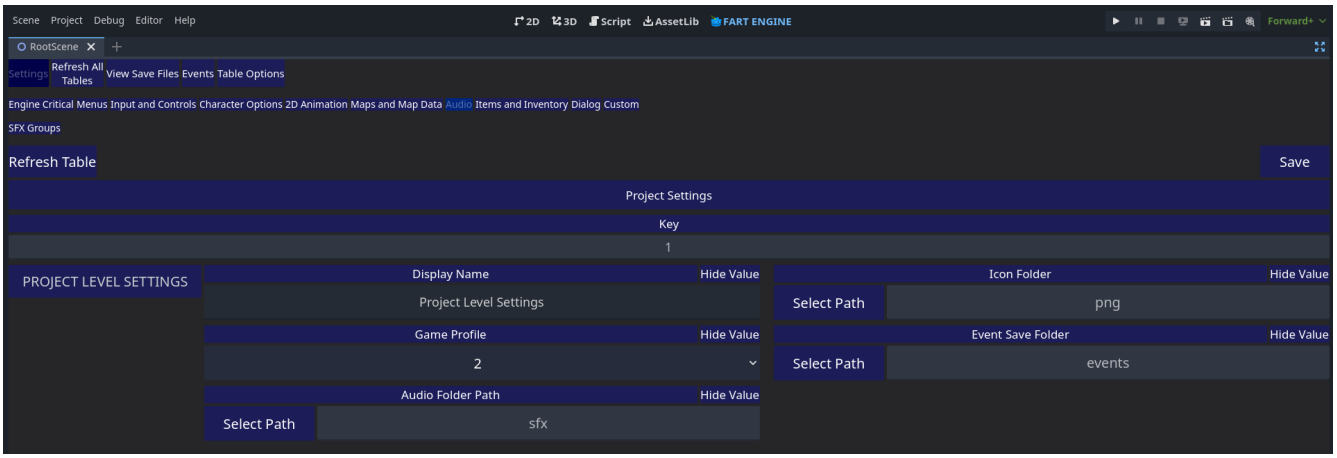
Show Value: If true, the field will show the current value, if false the Field Name Label will still be visible but the current value will be hidden

Tables: Reference table for Dropdown Data Type. Not available for any other Data Type

Accept: Apply changes and close the form

Cancel: Close form without applying changes

Settings



Project Level Settings

Game Profile: *Engine Critical → Global Data Table***: Tells the engine which game settings profile to use when compiled

*There are other options available but currently do nothing
**Path to Table - *Table Category → Table Name*

Table Options

Table Settings

Include in Save File: The engine has two in-game table categories. Static, which contains all of the tables. It is used in-game as read only. The other is Dynamic, which are the tables that will be saved and loaded with each save file. If you want a table to be included in Dynamic, set this option to true

Create Tab: If false, engine will not create a Table Button when the table's category is selected

Can Delete: If false, engine will not allow selected table to be deleted

Is Event: Designates table as an event. Likely no longer needed

Use Display Name in Editor: If true, the engine will use the display name instead of key value in the editor

Table Category: The category that selected table is assigned to

Show in Dropdown List: If true, selected table will show in Dropdown list of tables (Specifically for selecting a Reference Table for a Dropdown list)

Show Delete Table Button: Only one table needs this set to true, **Table Options**. Please do not change this unless you want to cause errors

Enable Key Options: When true, the Key Options in the Table Display can be accessed by user, when false, the options are not displayed. Field Options are Add Key, Delete Key, and Duplicate Key

Enable Field Options: When true, Field Options in the Table Display can be accessed by the user, when false, the options are not displayed. Field Options are New Field and Delete Field

Allow Duplicate Key Name: When true, table will allow user to have multiple Key's Display Name to be the same, when false, any duplicates will remain but engine will not allow any new duplicate Key's Display Name

Include in Event Conditions: When true, selected table will be populated in the list of tables available in the Conditions of Events

Table Buttons

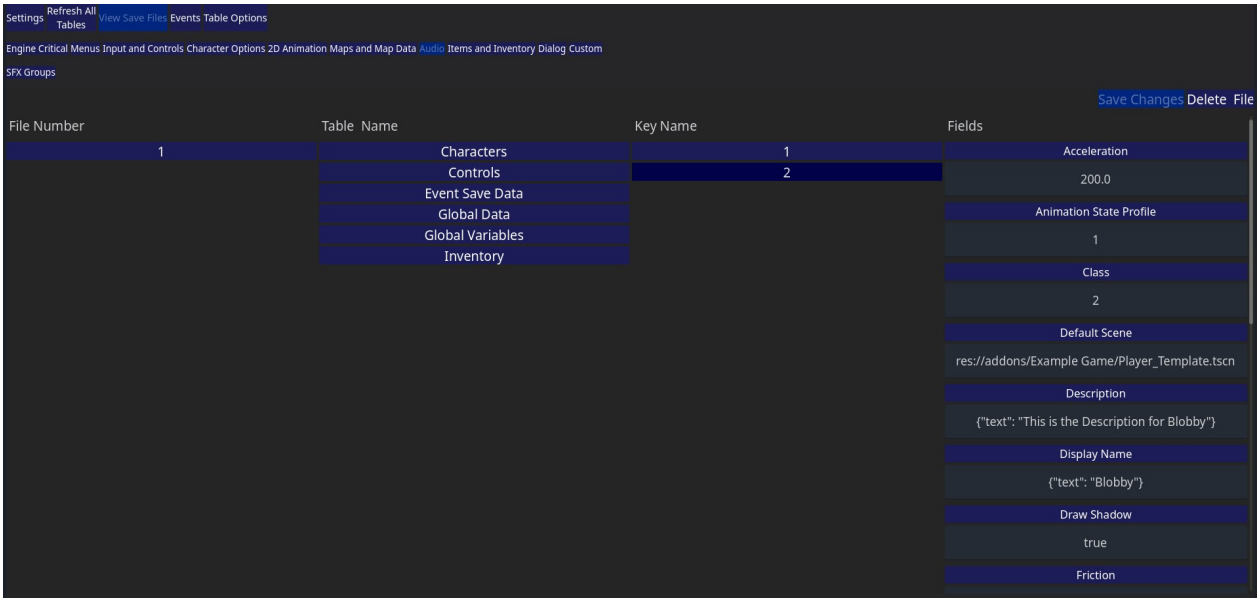
New Table: Adds a new table to the engine. See Table Options for New Table Form options

Delete Table: Deletes selected table. This is permanent and there is no popup warning or confirmation. Use with Extreme Caution

Save: Saves the options for the selected table and updates the engine

View Save Files

Allows user to view and edit the Dynamic tables saved when the player saves their game.
*The in-game save files are stored in the project's user:// folder. The file has a .sav extension and the data is stored in JSON format.



Save File Headers

File Number: Name of save file. Select file you want to see more information

Table Name: The tables that are available to view. Select a table to view the Keys

Key Name: Keys in the selected table. Select a Key view the Fields and Values

Fields: The Fields in the selected key. Select the Value to edit

View Save File Buttons

Save: Saves changes made to the field values of selected File/Table/Key. Use cautions when changing values in a save file

Delete File: Deletes the selected save file. If no save files are selected, the first file in the list is deleted

Characters

Character Options → Characters

Character Profiles

In Editor: Stores and creates playable characters and sets initial stats
In Game: Keeps track of character attributes after being added to the game, stored in Dynamic dictionary

Character Options

Max Speed: Maximum velocity in pixels per second that the character can move

Acceleration: How quickly character will reach Max Speed (Higher value is faster acceleration)

Friction: How quickly character will slow down before stopping

Mass: Combined with Global Gravity to set the velocity that character will fall. Only applied if Game Data table Is Gravity True, is true

Jump Speed: Combined with Global Gravity and mass to determine the initial jump velocity

Default Scene: Base **Godot** scene used to build the character in-game. Changing this could cause the character to display and behave incorrectly in-game

Animation State Profile: Animation Profile character will use in-game. Profiles are set in the Animation States table

Items

Items and Inventory → Items
In Editor: Stores and Creates Items and sets initial stats
In Game: Keeps track of Items after being added to the game, stored in Dynamic dictionary with table name of Inventory
*When a new game is created, all Items are added to player Inventory with ItemCount of 0

Item Options

ItemCount: The number of the selected Item the player has in Inventory. Should be 0 in editor

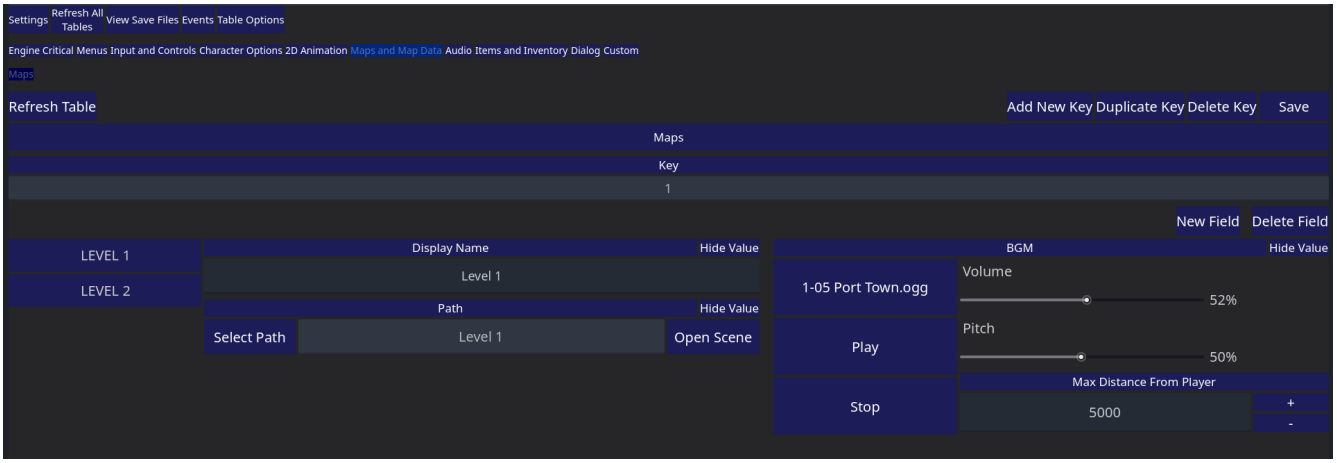
Maps

Maps and Map Data→ Maps

In Editor: Stores and designates variables for game Maps

In Game: Keeps track of Maps after being added to the game, stored in Dynamic dictionary

*The map scenes must be created using **Godot**



Map Options

Path: The path to the .TSCN file for the selected Map. The engine uses this to load the map in-game

BGM: In editor this value is used to set the starting BGM when scene is loaded

Event Editor

Events are containers for script that allows the character to interact with the in-game world. Responsible for handling all of the non-menu game logic. Used for everything from collectible objects to NPC

Event Page

Each Event has at least one. Each has the same input options. The engine will only load data for a single one at a time, called the Active Page. The Active Page is determined by checking the *Conditions*. Starting with the last Event Page (highest number) the engine checks Conditions. The first that returns true will be set to Active Page until the engine is prompted to check again

Event Page Buttons

Save Changes: Stores all data for the selected Event Page

Add New Event: Creates a new Event with default values

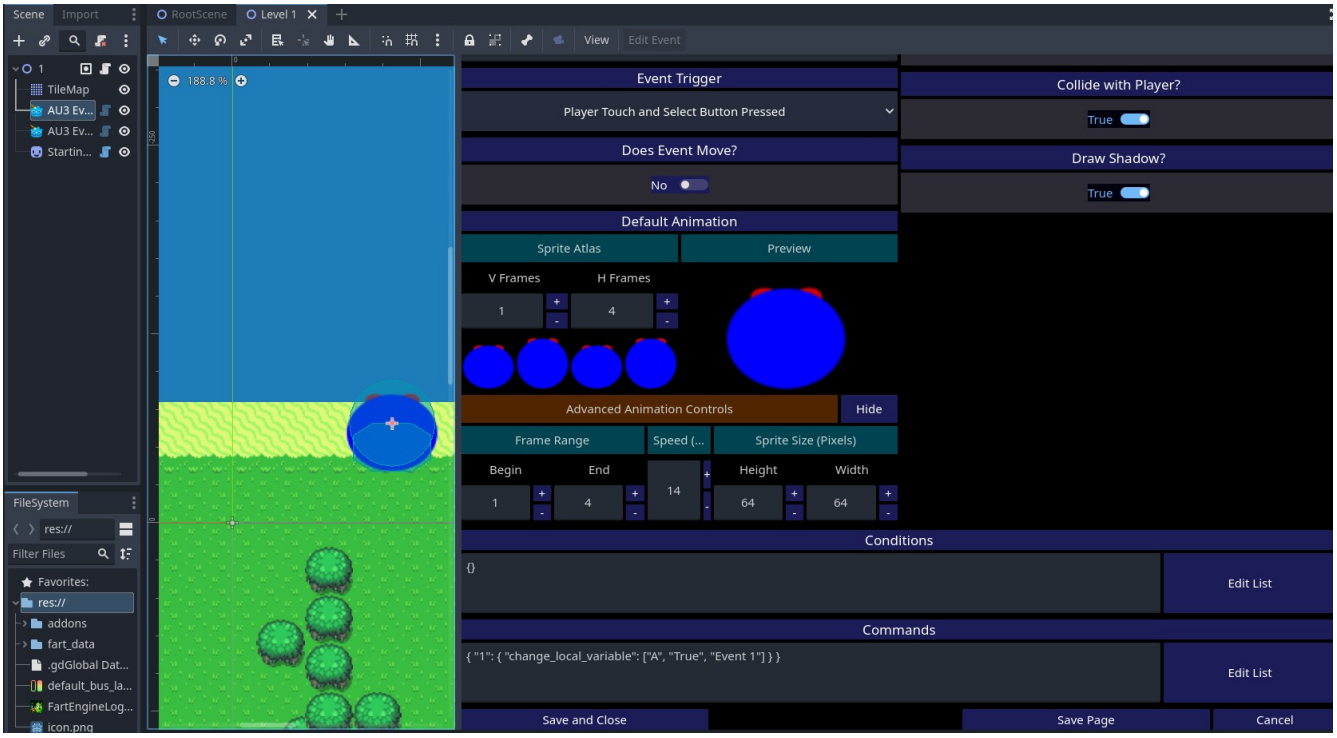
Delete Event: Removes the selected Event. Editor will prompt before removal

Add Page: Creates a new Event Page within the selected Event. The page number is one more than the largest page number

Duplicate Page: Creates a new Event Page using data from the selected Event. The new page is added to the end of the page list

Delete Page: Removes the selected Event Page. Disabled if Event has only one Page

Reset Values: Refreshes the Event Editor without saving



Save and Close: Stores all data for the selected Page and closes the Event Editor *

Save Page: Stores all data for the selected Page*

Cancel: Close Event Editor. Does not save any changes*

*Only available when accessing the Event Editor from Fart Event Node in the **Godot Map Editor**

Event Page Fields

Notes: Information for anyone working in the editor. Not used in-game

Event Trigger: The actions that must happen before the engine will run the command sequence.

Player Touch: When the player character touches this Event's Interaction Area

Player Touch and Select Button pressed: Player Touch and Interact button is pressed

Immediately: As soon as the event is loaded

Loop while Event Is Active: Immediately and repeatedly until event is removed or scene is changed

When Event Touches Another Event: When this event's Interaction Area touches another Event's Interaction Area

Does Event Move: If true, this event will be able to randomly move around the map. New options will appear in the Event Editor

Collide With Player: If true, the Event will get a collision shape and will act as a barrier the player cannot pass through

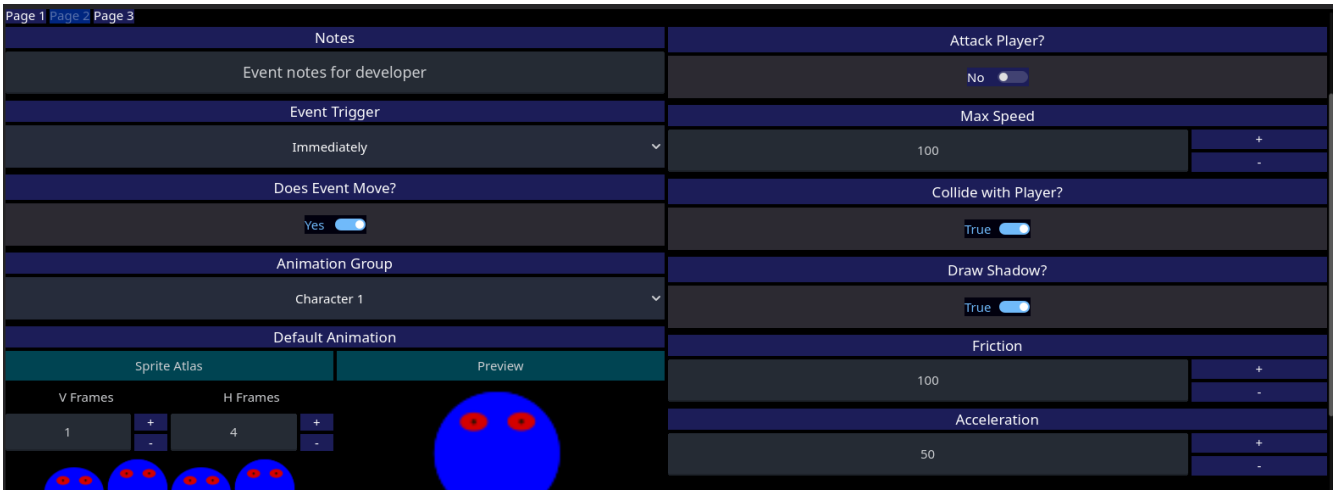
Draw Shadow: If true, will display a shadow based on the Sprite Animation or Animation Group

Conditions: Set the criteria for this Event Page to be active in-game*

Commands: Scripts that will run if this Page is active and the trigger is true*

*See Condition and Command sections for more detail

If Event CAN move



Animation Group: The set of sprite animations that will be applied to this event

*Options are from 2D Animation → Animation States table

Attack Player: If true, the event will chase the player if the Attack Player area is entered by player

Max Speed: Maximum velocity in pixels per second that the Event can move

Acceleration: How quickly Event will reach Max Speed*
*Higher value is faster acceleration

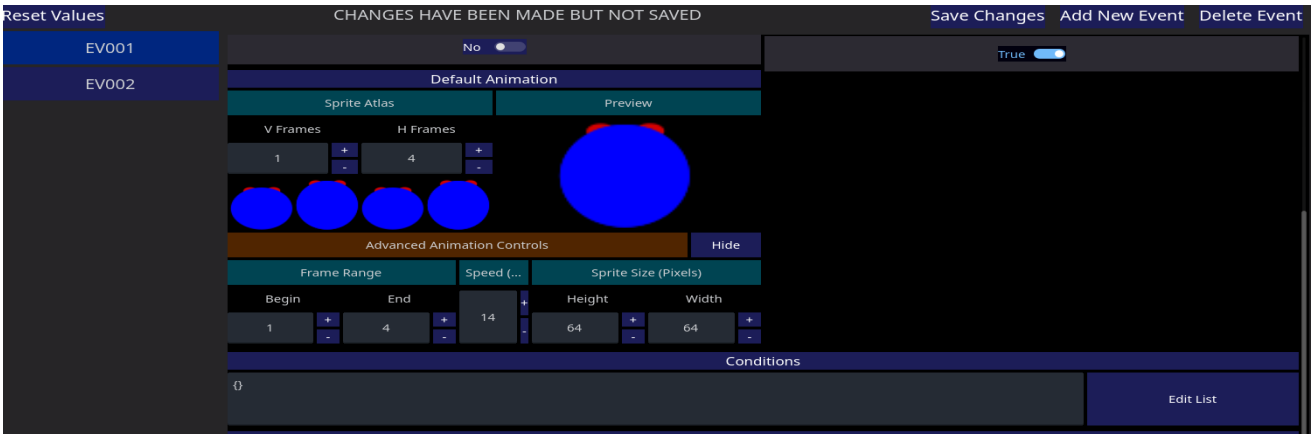
Friction: How quickly Event will slow down before stopping

If Event CANNOT move

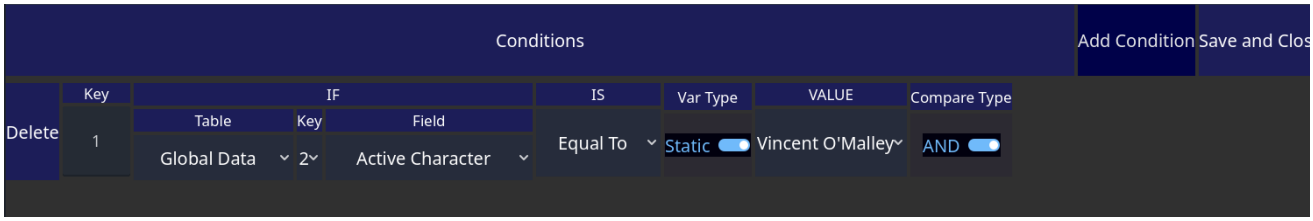
Loop Animation: if true, will play the selected Sprite Animation repeatedly until the Event is removed, the scene changes, or the Event Page changes

Conditions

Used to determine which Event Page is active



Conditions Editor

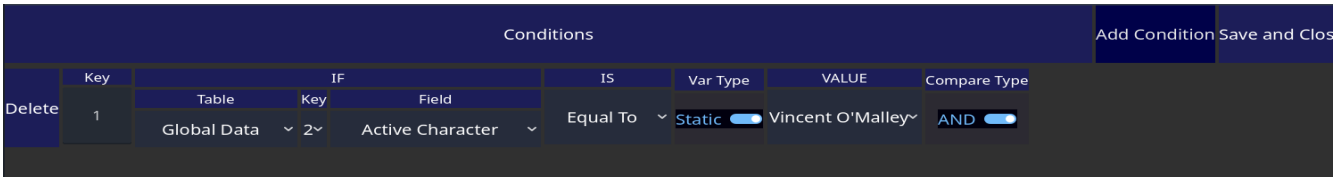


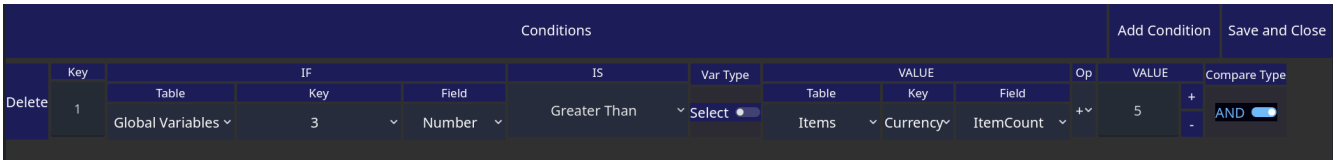
Add Condition: Creates a Condition Line Item at the end of the Conditions List

Save and Close: Updates the Condition Input Node. This does NOT save the Event Page

Delete: Removes the Condition Line Item

Condition Line Item





IF: Left side of equation

Table: Sets the current Table

Key: Sets the current Key

Field: Sets the current Field that contains the value the engine will use to compare to Right Side Value

IS: Comparison. Data Type of left side IF field will determine which comparisons are available. Numbers can use all options, text can only use “equals” or “does not equal”

Var Type: How the user will select the Right Side Field

Static: User will directly input desired Right Side Value (or select from Dropdown if Left Side Data Type is Dropdown list)

Select: User will select the Field from a list of tables in the engine that contain fields with only the same Data Type

Value1: Right Side of equation. See IF for more details

Op: What to do with Value1 and Value2 before comparing Left and Right values*

Value2: Extra static input for number values*

*Only available if Left Side Field Datatype is a number

Compare Type: Each line item tells the engine if all line items must return true, or if only one Line item needs to be true

AND: All line items that are set to AND must return true for Conditions to return true

OR: Only one OR line item needs to return true for Conditions to return true

Special Conditions Tables

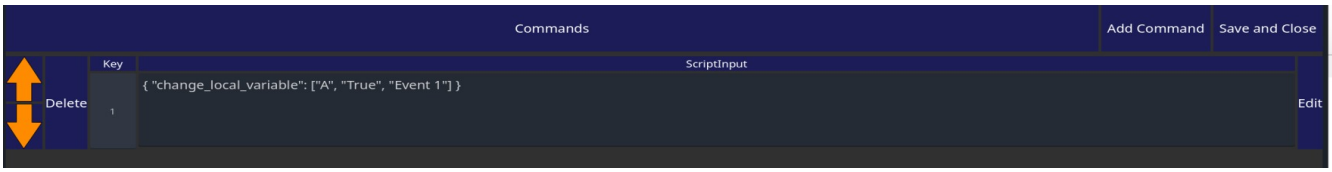
Items: Referenced in-game as Inventory

Global Variables: *Engine Critical → Global Variables*: Set at the game level, so all Events can access and change values

Local Variables: *Engine Critical → Local Variables*: Set at Event level. Can be used by all Event Pages but cannot be accessed outside of the Event

Commands

Actions that run when an Event Page is active and the trigger occurs. Controls the progression of your game



Command Editor

Script Output: The raw data that the engine uses to run Event Scripts

Add Command: Creates a new Command Line at the end of Command List

Save and Close: Stores to script data in the Command Node in Event Editor. The Event Page still needs to be saved for changes to persist

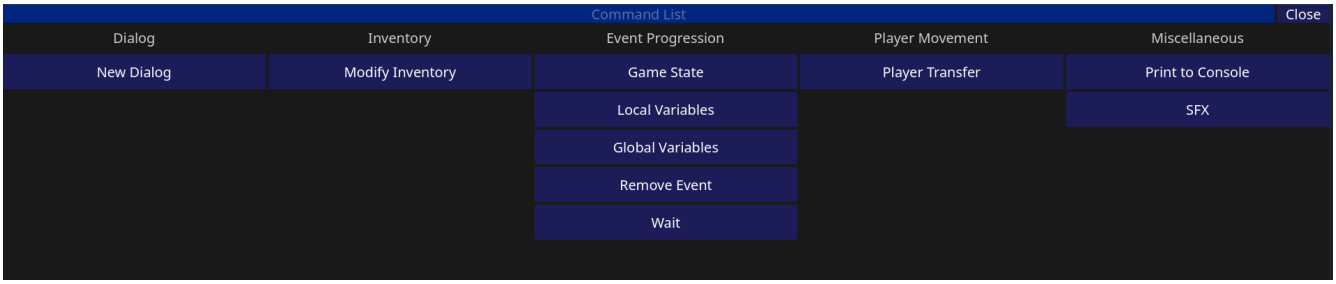
Command Line Item: Individual Commands listed in the order the engine will call them

Up/Down Arrows: Move the Command Line Item up or down in the Command List

Delete: Remove Command Line Item from Command List*
*Change is not permanent until Event Page is saved

Edit: Modify the Command Line item

List of Commands



Dialog

New Dialog: Adds a Dialog Command to the Command list

Inventory

Modify Inventory: *Items and Inventory → Items*: Add or remove Items from player Inventory

Event Progression

Game State: *Engine Critical → Game State*: Set the current Game State

Local Variables: *Engine Critical → Local Variables*: Change values that are accessible only to this Event
Global Variables: *Engine Critical → Global Variables*: Change values for variables that can be accessed by all Events
Remove Event: Delete the Event from the Current Map*
*Not persistent, must re-delete every time Event is loaded
Wait: Pause this Event from processing for a specified amount of time in seconds

Player Movement

Player Transfer: Move player to a different location, can be in Current Map or another selected from the Maps table

Misc

Print to Console: For debugging only. Prints a message to the **Godot Console Window**

SFX: Play a sound effect

Global Data

Contains Default settings for game user can set multiple profiles with different values for quicker testing. The Game Profile is set in Project Settings

Global Data Options

Starting Map: Sets the Map that will be used when player starts a new game. When a different Map is selected, the engine will open the new map and place a Starting Position Node. When the node is moved to a new location in the Map, the engine will update the Player Starting Position value

Player Starting Position: The vector coordinates the engine will use to set the player position when a new game is started. Can be set manually or by moving the Player Starting Position in the Starting Map

Game Title: The name of the game. Also, the text that will display in the Title Menu text label

Starting Character: *Character Options → Character*: Character that is used when player starts a new game

Game State: *Engine Critical → Game State*: Sets the state of the game when the project first starts

Is Gravity Active: If true, Engine will apply a downward (Y) force to player

Gravity Force: The amount of downward force is applied to player if Gravity is Active

Title Screen: Title Menu for the profile. Scene that loads after splash screen

Default Dialog Box: Default Dialog Scene. Can be overridden in Dialog Node

Default GUI: Scene that draws on top of the Game Window and contains all in-game menus

Default In-Game Menu: Base scene for navigating all in-game menus

Loading Screen: Scene that will show when moving to a new scene

Default Player Scene: Base character scene for the player

Options Only Used In-Game

Time: Date and time game was saved

Player POS: Global position of player

Current Map: Map that is active when game is saved

Is Game Active: If true, engine will not run scripts for New Game

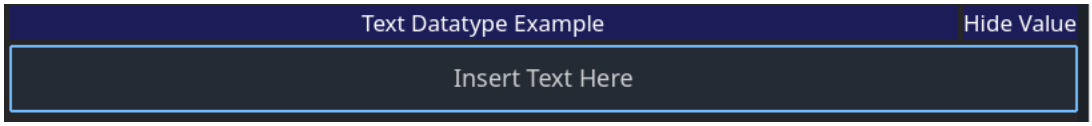
Active Character: Character theplayer is controlling when the game is saved

SaveID: Game Save ID

Fart Data Types

Text

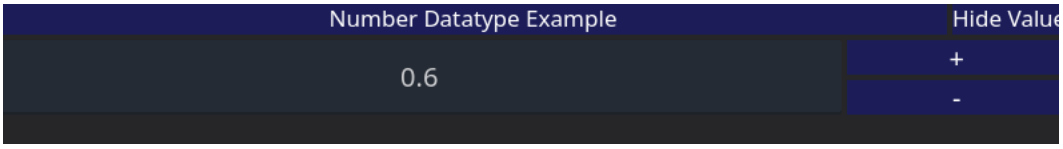
A string of any characters



1) Text to display

Number

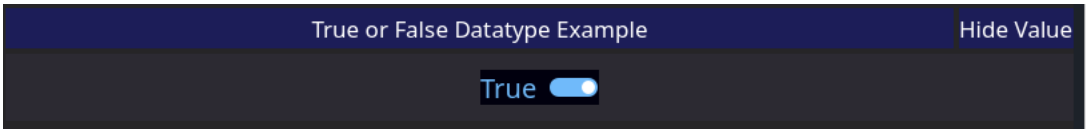
Used to store integers, negative numbers, and floating point (decimal) values



1) Number to be stored
2) Increase (1) by 0.1
3) Decrease (1) by 0.1

True or False

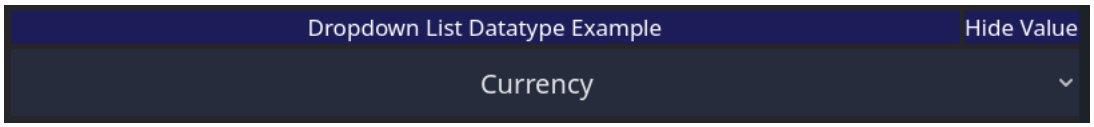
Stores boolean values

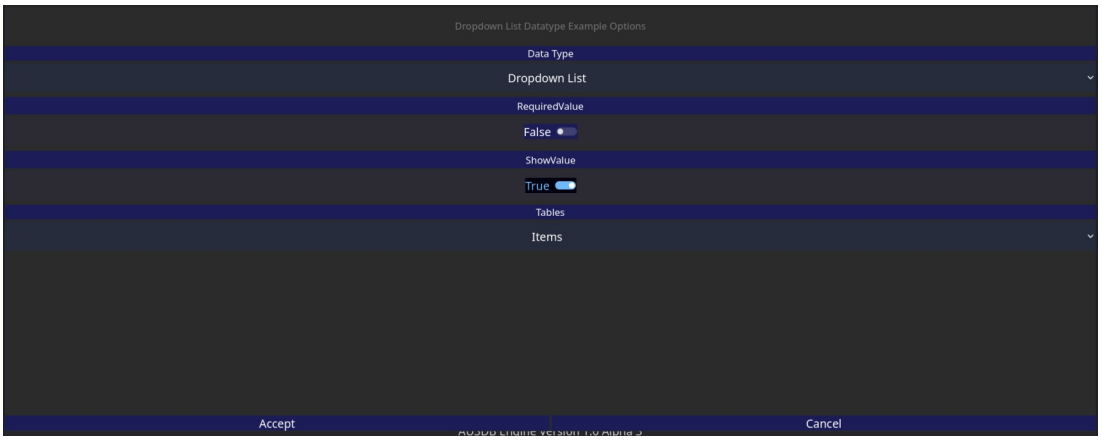


1) True/False toggle button

Dropdown List

Used to display and select a single Key from a table. The base table is set in the Field Options





1) **Key selection**: Select option to assign the key as the value

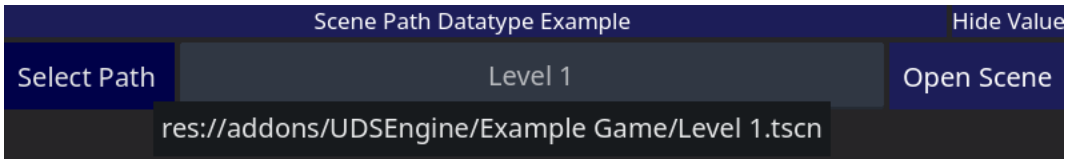
Icon Display

Used to display PNG files. File can be from anywhere inside or outside of project but when selected, the file is copied to “res://fart_data/png folder”

1)Icon Preview/Edit Button. Select to change selected PNG file

Scene Path

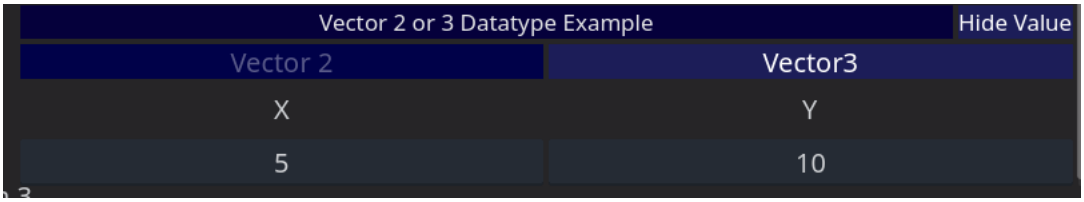
Used to store a path to a **Godot Scene**

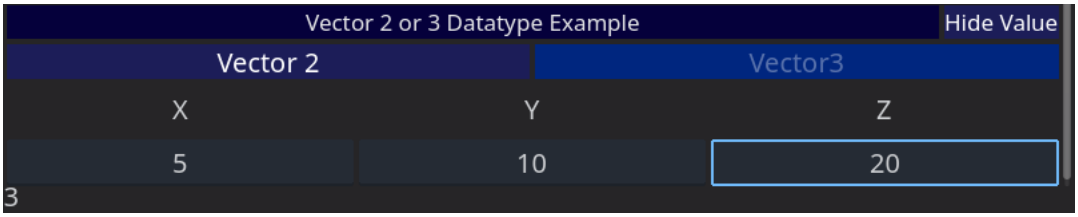


1)Scene selection/edit button. Select to edit selected scene
2)Scene Name (Cannot be edited directly). Hover over to see full path
3)Open selected scene in **Godot Editor**

Vector 2 or 3

Used to store a vector 2 or 3 value (X,Y) or (X,Y,Z). Must be numbers

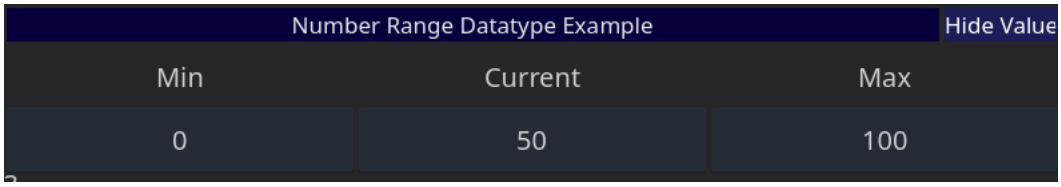




1) Vector 2 or 3 selection, change input vector type
2) X,Y,Z inputs, must be real numbers (positive or negative)

Number Range

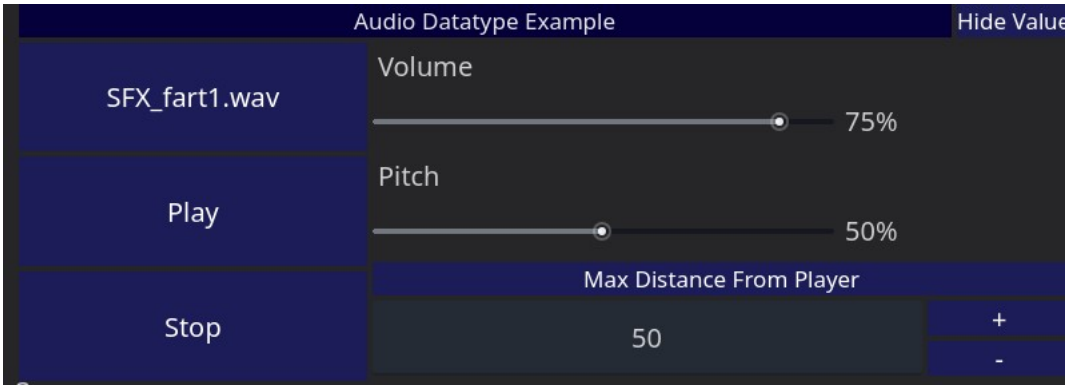
Used for storing a min, current, and max value. All must be real numbers



1) Min value: must be less than Max value
2) Current value: Must be between or equal to Min and Max values
3) Max value: Must be more than Min value

Audio

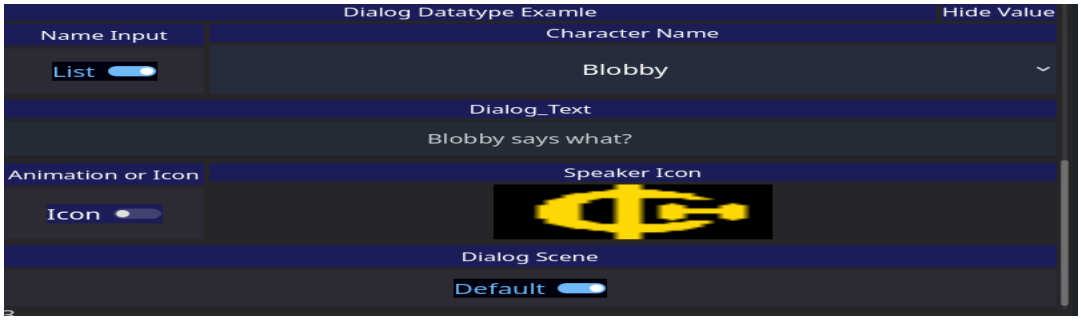
Used to store SFX or BGM paths and options



1) Displays audio file name. Select to change selected audio file
2) Play the selected audio file one time
3) Stops playback of selected audio file
4) **Volume Slider**: sets the default decibel output of the Audio file
5) **Pitch Slider**: Sets default pitch of the Audio file
6) Max distance (in pixels) the player can be away from source of sound before sound can't be heard*
   *Only applies to SFX in Events and not Map BGM

Dialog

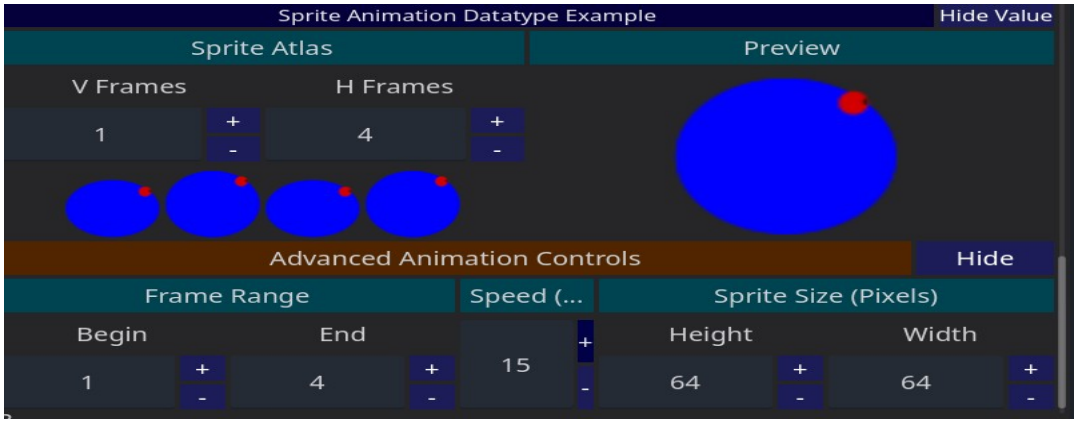
Used to create a single dialog instance in-game



1) **Text or Character Selection Checkbox**: User can manually type in speaker name or select from list of Characters
2) Dialog text
3) Speaker Icon or Sprite Animation selection
4) Icon or Sprite Animation data input
5) Default or Custom Dialog Scene Selection
6) Custom Dialog scene path selection

Sprite Animation

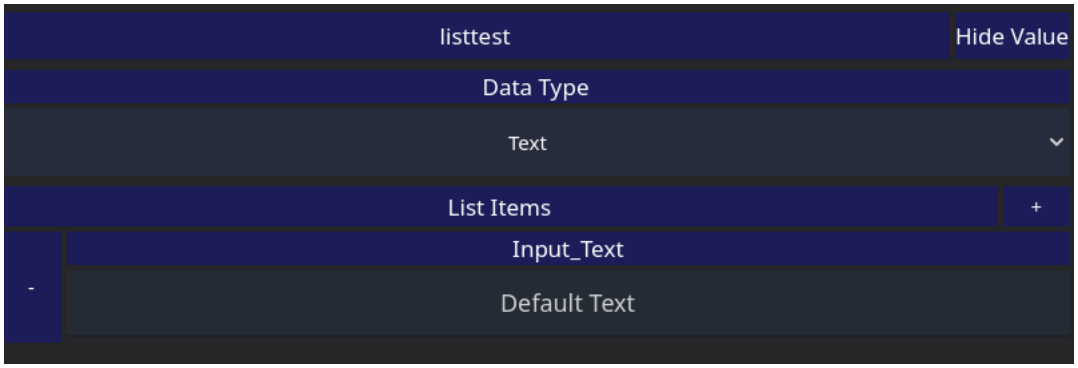
Stores all the necessary data to display a sprite animation in-game



1) **Sprite Map preview and Edit button**: Select to change selected sprite map
2) **Atlas vertical and horizontal frames**: The total vertical and horizontal frames of the sprite map
3) **Frame Range**: sprite map frames used in animation
4) **Speed**: Playback speed of the animation in frames per second
5) **Sprite Size**: Automatically sets based on sprite map size and number of frames but can be modified to any size
6) **Sprite Preview**: shows what the sprite animation looks like with options applied. Does not apply sprite size

List

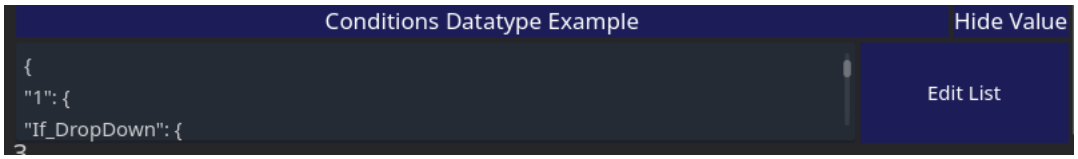
Stores a list of values with the same Data Type



1) **Data Type**: The datatype that will be used for all items in the list
2) Add item to list
3) Delete Item from list

Condition

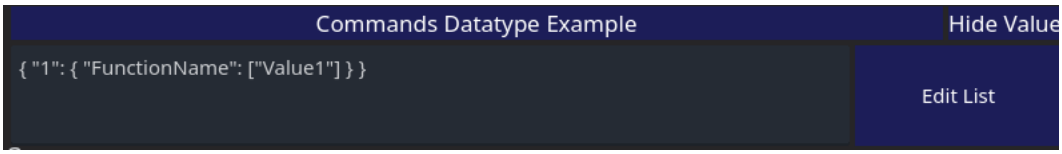
Only used inside Events. See Event for more details



1) Data Display: Shows the data that the engine uses in dictionary format
2) Edit Button: Open the form to edit the data from Data Display

Command

Only used inside Events. See Event for more details



1) **Data Display**: Shows the data that the engine uses in dictionary format
2) **Edit Button**: Open the form to edit the data from Data Display