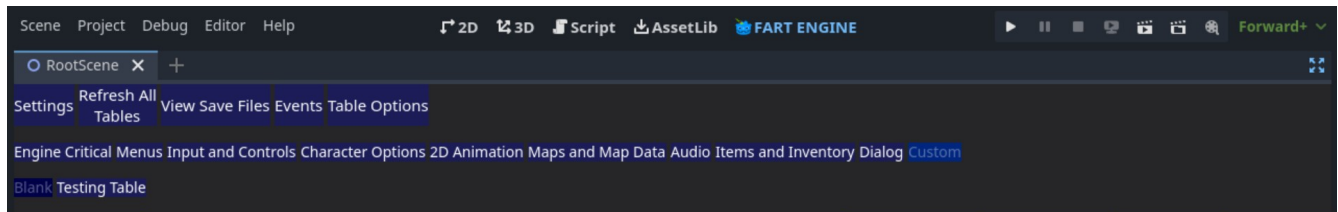


# Fart Engine Documentation

## Main Screen Navigation



### ***Top Row***

Contains engine top level options

**Refresh All Tables:** Reloads the Fart Engine display and data without saving 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

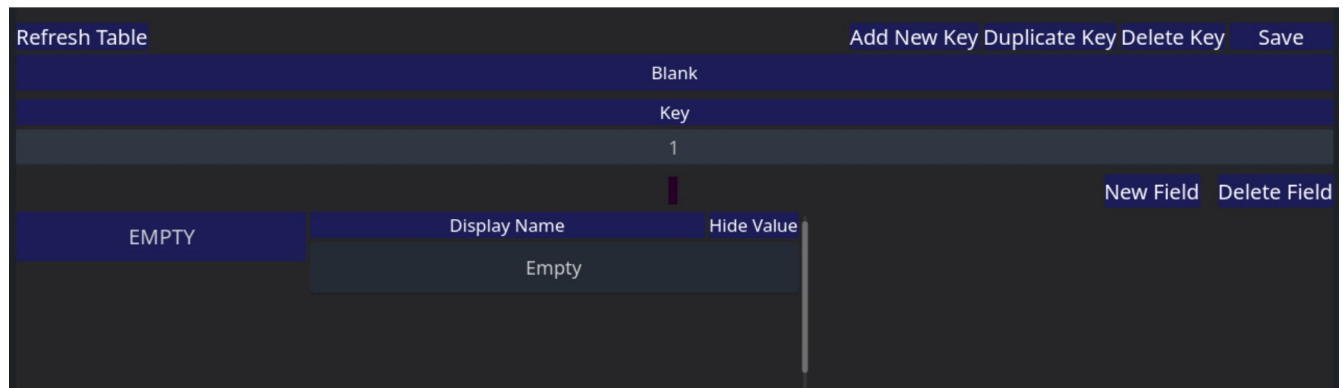
### ***Middle Row***

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

## Table 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 in 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 current value, 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 is set to use display name in the editor, the engine will use this value instead of the selected primary key

### ***How to rearrange 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 button 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

**Datatype:** Allows user to modify the datatype of selected field. Changing this will reset the selected field’s value to default for new datatype

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

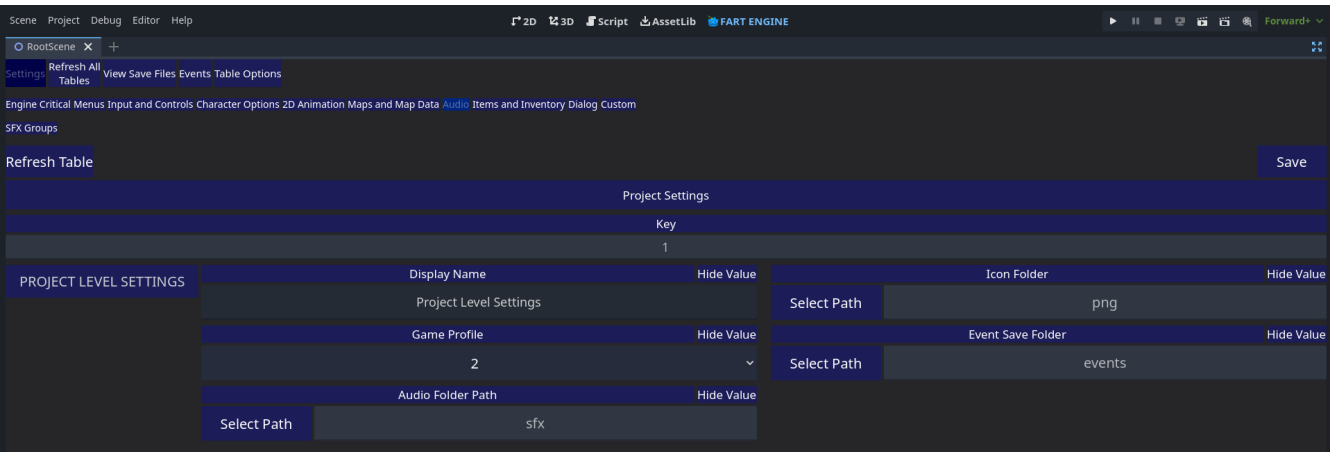
**Show Value:** If true, the field will show the current value, if false the Field Name Label will still be visible but the current value will be hidden

**Tables:** Reference table for dropdown Datatype. Not available for any other Datatype

**Accept:** Apply changes and close the form

**Cancel:** Close form without applying changes

# Settings: Project level settings

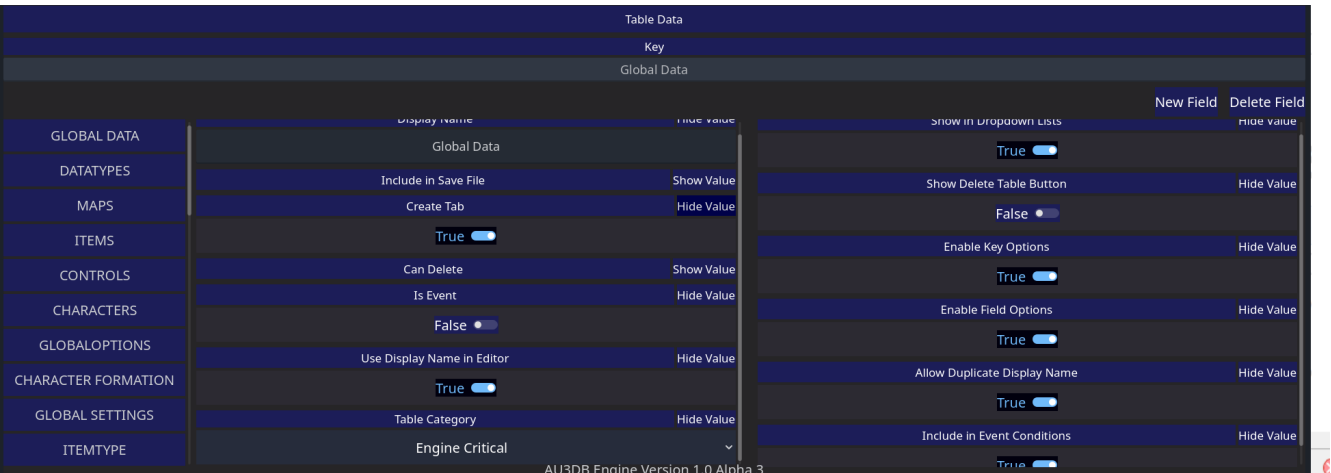


**Game Profile:** Engine Critical → Global Data Table: Tells the engine which game settings profile to use when compiled

\*There are other options available but currently do nothing

## Table Options

### Individual Table settings



**Include in Save File:** The Engine has 2 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 lists of tables (Specifically for selecting a “Reference Table” for a dropdown list

**Show “Delete Table” Button:** Only 1 table needs this set to true, The 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 (Options are, Add Key, Delete Key, and Duplicate Key) can be accessed by user, when false, the options are not displayed

**Enable Field Options:** When true, “Field Options” in the Table Display (Options are, New Field, Delete Field) can be accessed by the user, when false, the options are not displayed

**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

## ***Buttons***

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

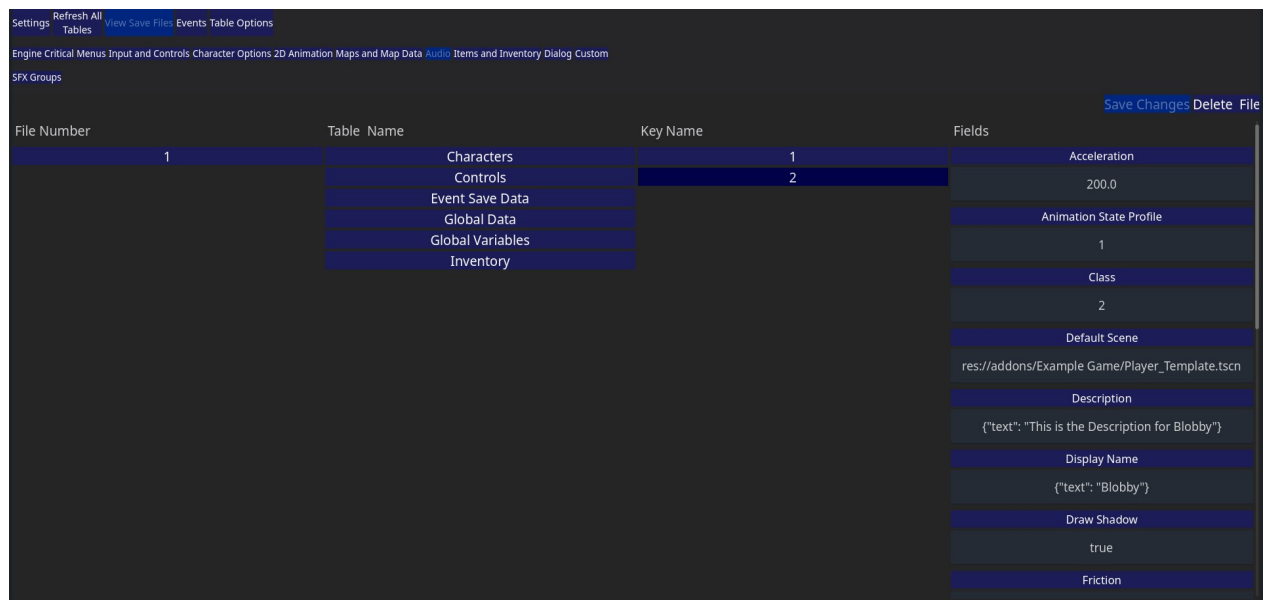
**Delete Table:** Deletes selected table. This is permanent and there is no popup warning or confirmation. Use with Extreme Caution

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

## **View Save Files**

Allows devs to view and edit the “Dynamic” tables saved when the user 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.



**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

## 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

**Character Profiles:** Character Options → Characters

**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

## Fields

**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 → “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

Settings Refresh All Tables View Save Files Events Table Options				
Engine Critical Menus Input and Controls Character Options 2D Animation Maps and Map Data Audio Items and Inventory Dialog Custom				
Items				
Refresh Table				
Add New Key Duplicate Key Delete Key Save				
Items				
Key				
2				
				New Field Delete Field
CURRENCY	Type	Show Value	CanUse	Show Value
	CanEquip	Show Value	Description	Show Value
	Attack	Show Value	CanSell	Show Value
	Defense	Show Value	ItemCount	Hide Value
	Speed	Show Value	0	+
	HealthRecovery	Show Value		-
	Durability	Show Value	InventoryItem	Show Value
	SelfValue	Show Value	CanCarry	Show Value
	Cost	Show Value	Icon	Show Value
			IconDescription	Show Value
				Display Name Hide Value
Currency				

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

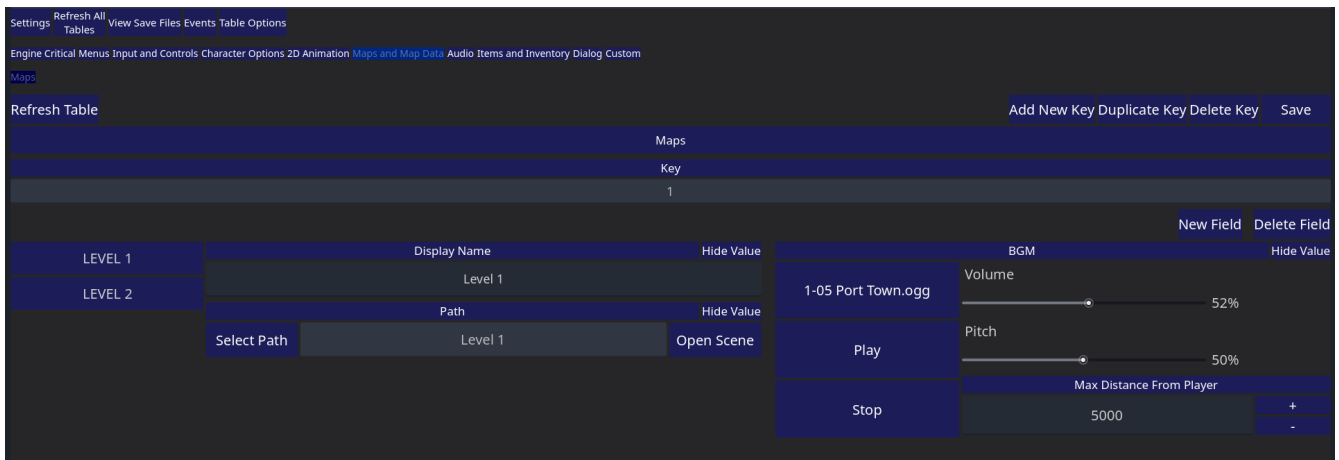
# Maps

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



**Path:** The path to the .tscn file for the selected map. The Engine uses this to load the map in-Game

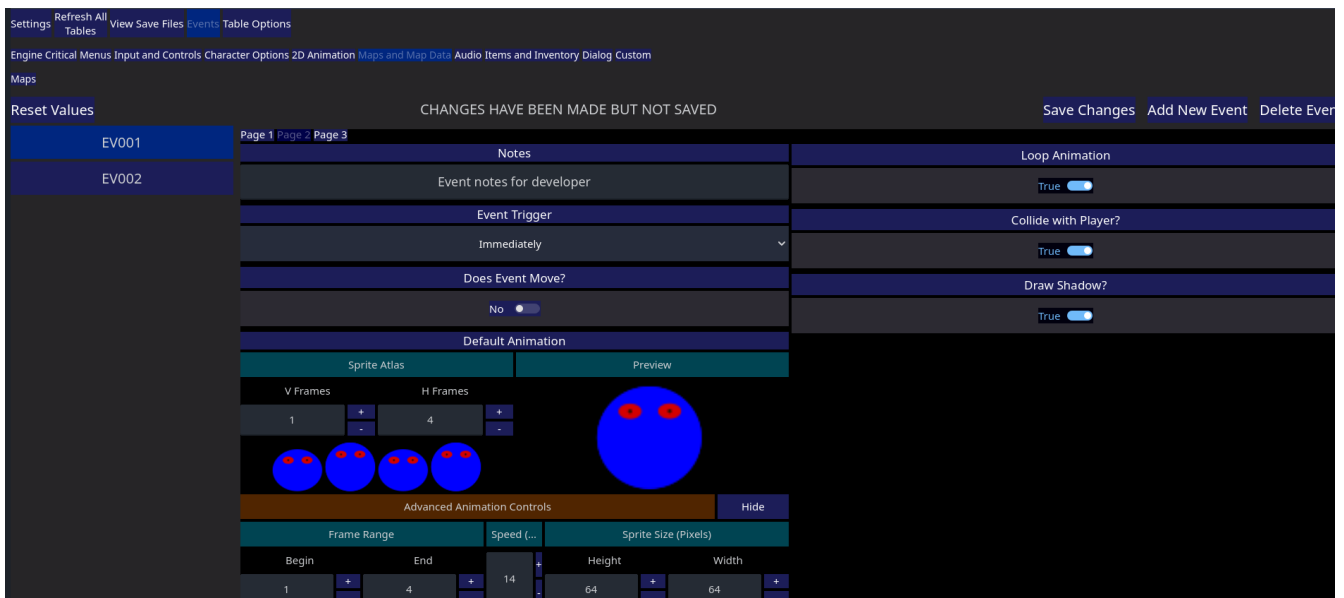
**BGM:** In editor this value is used to set the starting BGM when scene is loaded

## Event Editor

What are Events: 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 Input Fields

*Each page contains the following 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 shade 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***

The screenshot shows the Event Editor interface with the following settings:

- Page 1** | **Page 2** | **Page 3**
- Notes:** Event notes for developer
- Event Trigger:** Immediately
- Does Event Move?:** Yes
- Animation Group:** Character 1
- Default Animation:** Sprite Atlas
- Attack Player?:** No
- Max Speed:** 100
- Collide with Player?:** True
- Draw Shadow?:** True
- Friction:** 100
- Acceleration:** 50
- V Frames:** 1
- H Frames:** 4
- Preview:** A blue circular sprite with red eyes.

**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

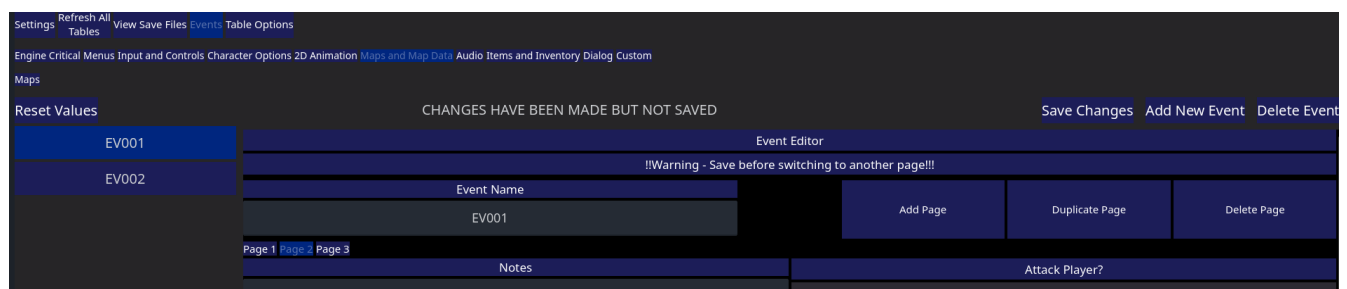
### ***If Event CANNOT move***



**Loop Animation:** if true, will play the selected “Sprite Animation” repeatedly until the Event is removed, the scene changes, or the Event page changes

### ***Pages***

Each Event has at least 1 Page. Each Page has the same input options. The Engine will only load data for a single Page at a time, called the Active Page. The Active page is determined by checking the Page Conditions. Starting with the last page (highest number) the Engine checks Conditions. The first Page that returns true will be set to Active until the Engine is prompted to check again



## Page Buttons

**Save Changes:** Stores all data for the selected Page

**Add New Event:** Creates a new Event with Default values

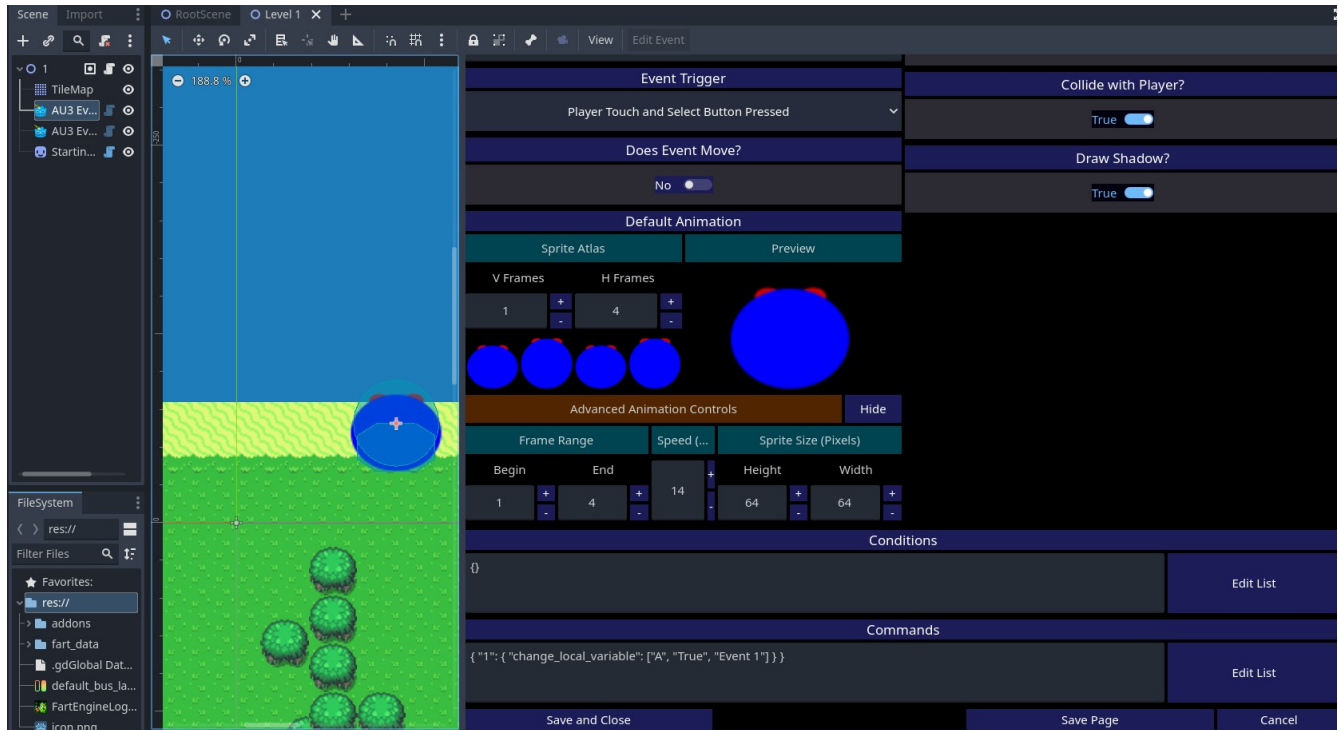
**Delete Event:** Removes the selected Event. Editor will prompt before removal

**Add Page:** Creates a new Page within the selected Event. The page number is 1 more than the largest Page number

**Duplicate Page:** Creates a new page using data from the Selected Event. New Page is added to the end of the Page list

**Delete Page:** Removes the selected Page. Disabled if Event has only 1 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 Map Editor

# Conditions

Used to determine which Event Page is Active

Reset Values

CHANGES HAVE BEEN MADE BUT NOT SAVED

Save ChangesAdd New EventDelete Event

EV001

No

True

EV002

Default Animation

Sprite Atlas

Preview

V Frames

H Frames

1

+

-

4

+

-

Advanced Animation Controls

Hide

Frame Range

Speed (...)

Sprite Size (Pixels)

Begin

End

14

1

+

-

4

+

-

64

+

-

64

+

-

Conditions

0

Edit List

## Conditions Editor

Conditions

Add ConditionSave and Close

Delete

Key

1

IF

Table

Key

Field

Global Data

2

Active Character

IS

Equal To

Var Type

Static

VALUE

Vincent O'Malley

Compare Type

AND

**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

Conditions

Add ConditionSave and Close

Delete

Key

1

IF

Table

Key

Field

Global Data

2

Active Character

IS

Equal To

Var Type

Static

VALUE

Vincent O'Malley

Compare Type

AND

Conditions										Add Condition	Save and Close		
Delete	Key	IF		IS	Var Type	VALUE			Op	VALUE	Compare Type		
	Table	Key	Field			Table	Key	Field					
	1	Global Variables ▾	3 ▾	Number ▾	Greater Than ▾	Select ●	Items ▾	Currency ▾	ItemCount ▾	+ ▾	5	+ ▾	AND <input checked="" type="checkbox"/>

Conditions										Add Condition	Save and Close	
Delete	Key	IF		IS	Var Type	VALUE			Op	VALUE	Compare Type	
	Table	Key	Field			Table	Key	Field				
	1	Global Variables ▾	3 ▾	Number ▾	Greater Than ▾	Static <input checked="" type="checkbox"/>	0	+ ▾	+ ▾	5	+ ▾	AND <input checked="" type="checkbox"/>

**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. Datatype 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”

**VarType:** 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 datatype is dropdown list)

**Select:** User will select the Field from a list of tables in the Engine that contain fields with only the same datatype

**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 1 Line item needs to be true

**AND:** All line items that end in AND must return true for Conditions to return true

**OR:** Only 1 OR line item needs to return true for Conditions to return true

## Special Conditions Tables

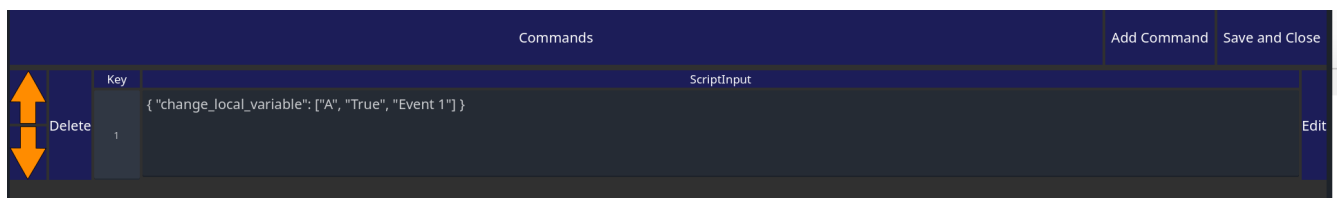
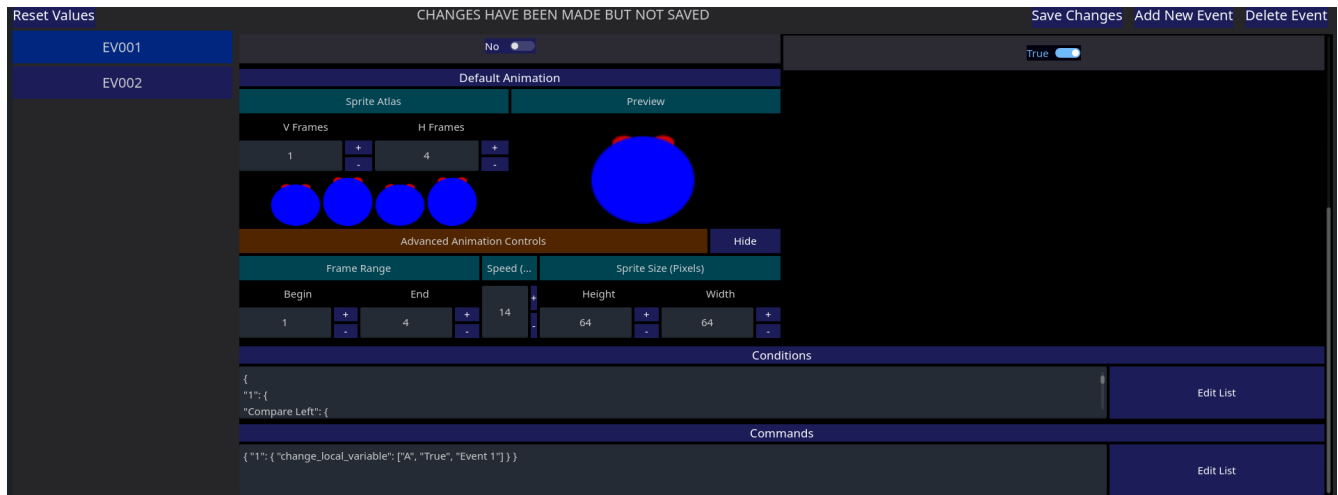
**Items:** In-Game will be referenced as “Inventory”

**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



**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 line item up or down in the Command List

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

**Edit:** Modify the Command Line item

# List of Commands

Command List					Close
Dialog	Inventory	Event Progression	Player Movement	Miscellaneous	
New Dialog	Modify Inventory	Game State	Player Transfer	Print to Console	
		Local Variables			
		Global Variables			
		Remove Event			
		Wait			
		SFX			

## Dialog

**New Dialog:** Adds a Dialog Command to the Command list

## Inventory

**Modify Inventory:** Items and Inventory → Items: Add or Remove Items from Player's Inventory

## Event Progression

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

**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 Command List 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 a different one

## Misc

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

**SFX:** Play a sound effect

# Global Data

Settings

Refresh All Tables

View Save Files

Events

Table Options

Engine Critical

Menus

Input and Controls

Character Options

2D Animation

Maps and Map Data

Audio

Items and Inventory

Dialog

Custom

Global Data

DataTypes

Global Settings

Event Table Template

Global Variables

Inequalities

Event Save Data

Game State

Table Category

Local Variables

Operations

UI Methods

Refresh Table

CHANGES HAVE BEEN MADE BUT NOT SAVED

Add New Key

Duplicate Key

Delete Key

Save

Global Data					
Key					
1					
				New Field	Delete Field
2	NewGame	Show Value	Is Gravity Active	Show Value	
	Starting Map	Show Value	Gravity Force	Show Value	
	Time	Show Value	Title Screen	Show Value	
	Player POS	Show Value	Default Dialog Box	Show Value	
	Current Map	Show Value	Default Menu Controls	Show Value	
	Game Title	Show Value	Default GUI	Show Value	
	Is Game Active	Show Value	Project Root Scene	Show Value	
	Starting Character	Show Value	Default In-Game Menu	Show Value	
	Active Character	Show Value	Loading Screen	Show Value	
	Player Starting Position	Show Value	Default Player Scene	Show Value	
	Default Controls	Show Value	Display Name	Show Value	
	Game State	Show Value	Save ID	Show Value	

## Contains Default settings for Game

User can set multiple profiles with different values for quicker testing. The Game Profile is set in Project Settings

**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” input

**Player Starting Position:** The vector coordinates the Engine uses 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 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 Game is started

**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 input 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's

### ***Fields 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 the Player is controlling when the game is saved

**SaveID:** Game Save ID

## **Fart Datatypes**

### ***Text***

A string of any characters

Text Datatype Example	Hide Value
Insert Text Here	

1) Text to displayed

### ***Number***

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

Number Datatype Example	Hide Value
0.6	+
	-

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

## ***True or False***

Stores Boolean values

True or False Datatype Example	Hide Value
True <input checked="" type="checkbox"/>	

- 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”

Dropdown List Datatype Example	Hide Value
Currency	

Dropdown List Datatype Example Options	
Data Type	Dropdown List
RequiredValue	False
ShowValue	True
Tables	Items
Accept	
Cancel	

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

## ***Icon Display***

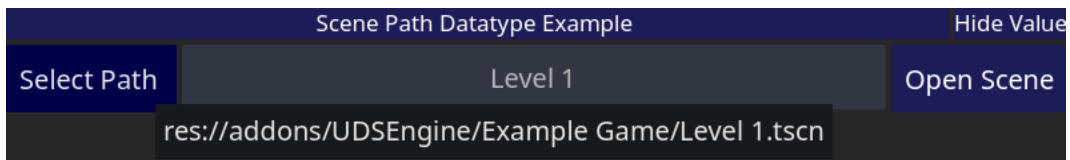
Used to display PNG files. File can be from anywhere inside or outside of project but when selected, the file is copied to “res://data/png folder”



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

## Scene Path

Used to store a path to a Godot scene



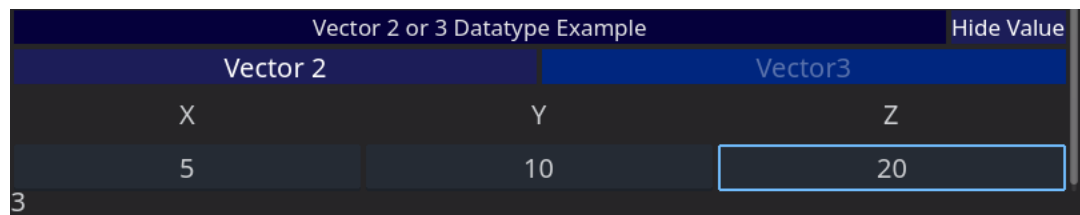
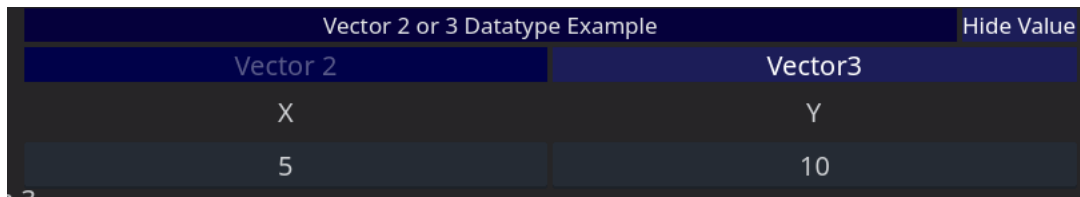
1)Scene selection/Edit button. Select to edit selected scene

2)Scene Name (Cannot be edited). 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

Number Range Datatype Example			Hide Value
Min	Current	Max	
0	50	100	

- 1) Minimum value: must be less than Max value
- 2) Current value: Must be between Minimum and Max values
- 3) Max value: Must be more than Minimum value

## Audio


Used to store SFX or BGM paths and options

Audio Datatype Example		Hide Value
SFX_fart1.wav	Volume	75%
Play	Pitch	50%
Stop	Max Distance From Player	50
		+
		-

- 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

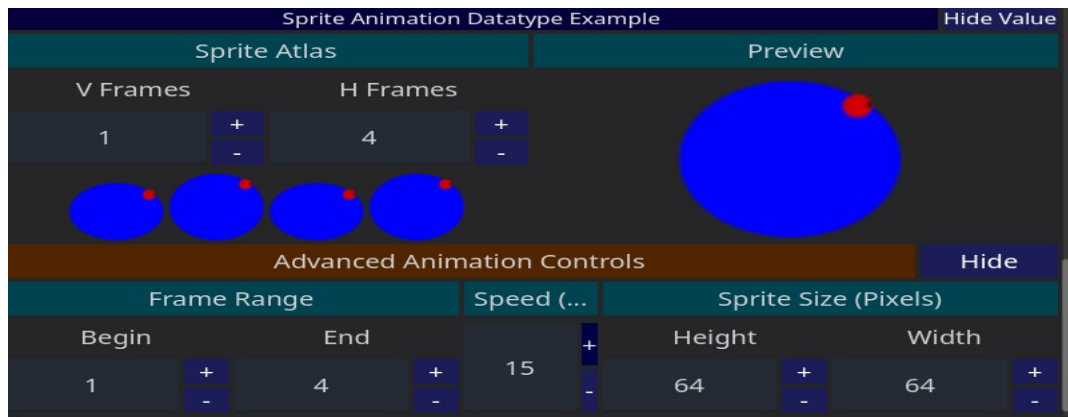
Used to create a single dialog instance in-game

Dialog Datatype Example		Hide Value
Name Input	Character Name	
List <input checked="" type="checkbox"/>	Blobby	▼
Dialog_Text		
Blobby says what?		
Animation or Icon	Speaker Icon	
Icon <input checked="" type="checkbox"/>		
Dialog Scene		
Default <input checked="" type="checkbox"/>		

- 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

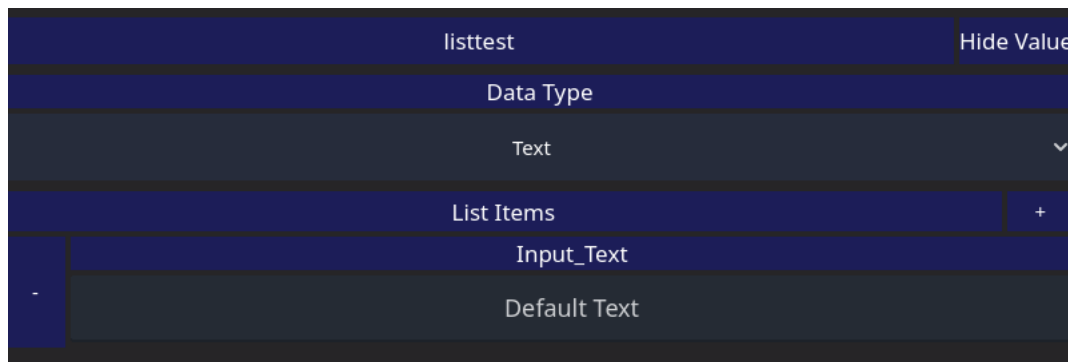
Stores all the necessary data to display a sprite animation in-game



- 1) Sprite Map preview and Edit button. Select to change selected Sprite map
- 2)Atlas Vertical and Horizontal coordinates. The total Vertical and Horizontal frames of the sprite map
- 3) Frame Range: sprite map frames used in animation
- 4)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

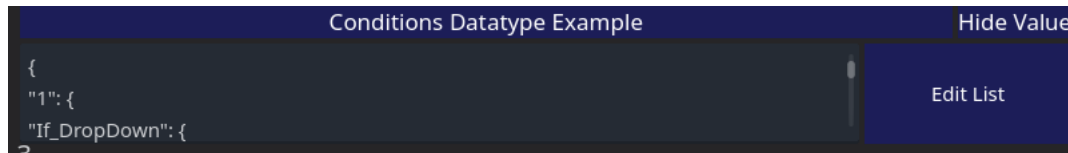
Stores a list of values with the same datatype



- 1) Datatype selection: The datatype that will be used for all items in the list
- 2) Add item to list
- 3) Delete Item from list

## **Condition**

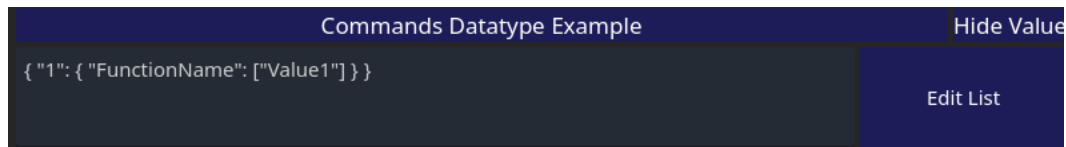
Only used inside Events. See Event Documentation for more details



- 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 Documentation 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