## MVDE:MM Visual Dialogue Editor

### About

MVDE is a custom editor window for the Unity editor created by MirMirror. It assists with creating, editing, storing dialogue trees and playing dialogue for use in games. MVDE provides a good solution to organize and play your dialogues in your project.

### Contents

## Set Up

### File placement

The package that MVDE comes in is very easy to set up.
Firstly, we recommend not moving the assets from their original folder. Otherwise, the software will not work properly.

To open the window in Unity Editor go to the Tools menu and open MVDE.

### Sample Scene

This asset comes with a sample scene. It can be found in the Demo folder as Demo. The scene will play through a dialogue tree in Game Scene. Once the scene has started click skip button to progress through the dialogue. If a choice is presented, click the corresponding item to choose your answer.
The dialogue is driven by MMDialogueController reading the scriptable object DemoDialogueData. MMDialogueController can be used to read other MMDialogue_Data objects.

### What is a Dialogue Tree?

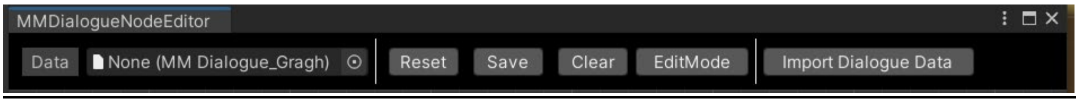
A dialogue tree is a data tree for storing dialogue. They are used to store dialogue which branches, this means the dialogue can go in mulitple directions based on conditions.
The most common form of branching dialogue is one where the player is asked a question and presented with multiple answers. The answer that the player gives will change what the next dialogue will say. In this case the condition for the branch is the player’s answer.

### What are the advantages of MVDE?

In addition to the visual dialogue editor, we also provide a unique rich text editor. You can set ruby, blood, italic, font size and typing speed in my RTF editor. If you have the speaker's voice-over, you can also set it in MM_DialogueNode.

## Controls

### Window Controls



*Object field* - This lets you select a scriptable object of the class MMDialogue_Gragh. If one is selected it will load from and save to this file.

*Button [Reset]* - This button will load data from the MMDialogue_Gragh file in the object field, if there isn't one selected it wont do anything.

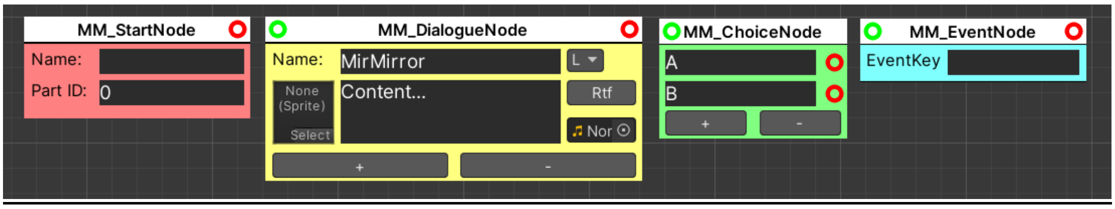
*Button [Save]* - This button will save the current data to a MMDialogue_Gragh file. If there is a file in the object field it will overwrite it. If there in no file selected it will open a file picker and you can choose where to save the file.

*Button [Clear]* - This button will clear all the workspace and only leave a MM_StartNode.

*Button [EditMode]* - This button will preprocess RTF or show RTF.

*Button [Import Dialogue Data]* - This button will save a new MMDialogue_Data file and a new MMDialogue_Gragh file.

### Node Controls



*Header* - Click and drag the node header to move the node. The header will darken while it is clicked. Right click in the node header, then you can select the delete item to delete the node.

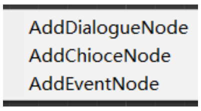
*Button [🔴]* - This button will activate the connection line. The line will be drawn to your mouse click on the top left *button[🟢]* of a node to connect the arrow to another node.

*Button [+]* - This button will add another section to a node.

*Button [-] - This button will remove a section from a node. Be aware that there is a minimum number of sections that a node must have.*

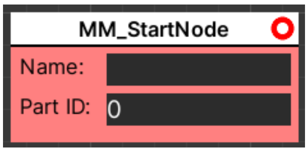
### Creating Nodes

Right click in the workspace will open the node menu. Click on either AddDialogueNode, AddChoiceNode or AddEventNode to create the corresponding node. Left clicking anywhere outside the menu will close it.



## Types of Nodes

### MM_StartNode

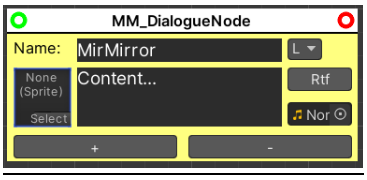


Name:

Part ID: 0

The start node is the first node in the tree. There are two purposes. The first one is to connect to a dialogue node to tell the code where the tree starts. Another one is to distinguish different MMDialogue_Data by Name and PartID.

### MM_DialogueNode

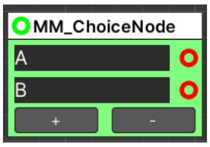


Name: MirMirror

Content...

The dialogue node holds the majority of text in the dialogue tree. You can set the speaker's name, images, contents, position on screen, voice-over and edit RTF by clicking button[RTF]. It starts with one text box but more can be added. The dialogue node can connect to other dialogue nodes, choice nodes and event nodes.

### MM_ChoiceNode

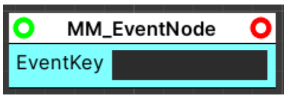


A

B

The choice node is what creates branches in the tree. It shows what the player's choices will be in the dialogue. It starts with two choices, "A" and "B", but they can be edited and more choices can be added. It can connect to as many nodes as there are choices.
Not every choice has to connect to a node. It can connect to dialogue and event nodes.

### MM_EventNode



EventKey

The event node is used to call UnityEvents in the dialogue tree. It only holds a text field labelled "Event Key". When the dialogue tree is parsed, you can use the event key to trigger a UnityEvent at the specified part of the tree. To do this, you would need to send a collection of events with the tree too.

## Saving & Loading

Dialogue trees are saved to MMDialogue_Gragh scriptable object files. DEW can also load the saved dialogue trees from MMDialogue_Gragh files.

MMDialogue_Gragh scriptable object file has an important scriptable object of the class MMDialogue_Data which will be used in MMDialogueController.

When saving data, having a MMDialogue_Gragh file in the object field will change what happens. If there is a file in the object field, the file's data will be overwritten with the new saved data.

If there is no file in the object field, the file picker will appear. If a file is overwritten in the file picker references inside Unity Editor, it will be lost.

## Contact

If you require assistance with setting up MVDE or have any queries about the tool, do not hesitate to contact us at MirMirror.

Email:MirMirror@outlook.com