# Narrative Tool Document

Dominic McNeill – 102061800

Updated for version – 1.0.0

Repository link - <https://github.com/domibron/NarrativeToolUnity>

# Table of Contents:

Contents

[Narrative Tool Document 1](#_Toc184310277)

[Table of Contents: 2](#_Toc184310278)

[Downloading and installing the tool: 3](#_Toc184310279)

[Installing inside Unity: 5](#_Toc184310280)

[Using the tool to create a dialog asset: 8](#_Toc184310281)

[Navigation: 10](#_Toc184310282)

[Creating a node: 11](#_Toc184310283)

[Breakdown of the node: (PUT UNIVERSE IMAGESSSS) 12](#_Toc184310284)

[Option display field: 12](#_Toc184310285)

[Input and Output: 13](#_Toc184310286)

[Dialog box: 16](#_Toc184310287)

[Start node: 17](#_Toc184310288)

[Tool bar: 18](#_Toc184310289)

[Saving: 18](#_Toc184310290)

[Note: 19](#_Toc184310291)

[Loading: 20](#_Toc184310292)

[Clear: 22](#_Toc184310293)

[Scriptable asset: 23](#_Toc184310294)

[Code: 24](#_Toc184310295)

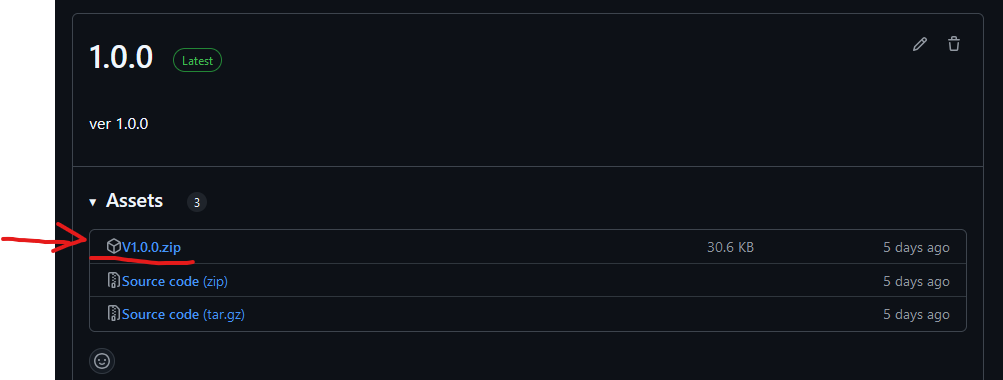
# Downloading and installing the tool:

You will need Unity. Ideally, Unity **2023.2**, please use this version as it is the tested version.

Please download the tool from the releases.

A screenshot of a computer

Description automatically generated



Note:

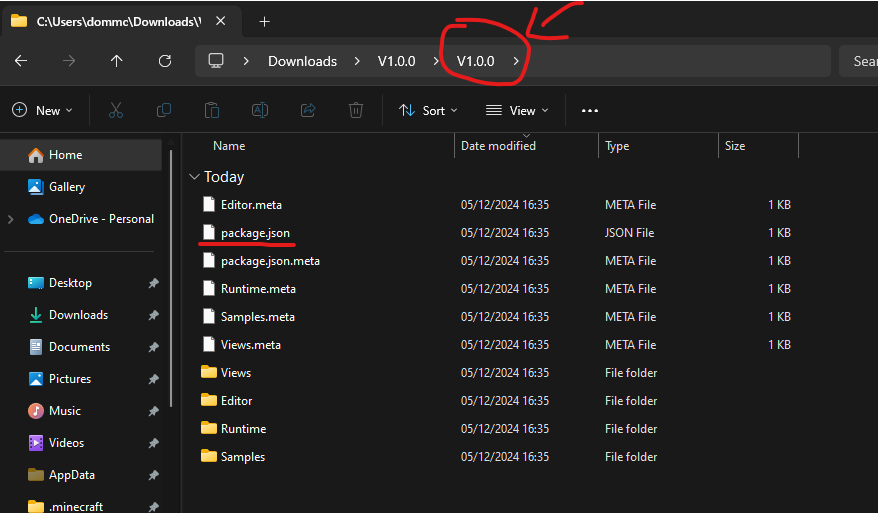
Make sure to extract the zip!

A screenshot of a computer

Description automatically generated

# Installing inside Unity:

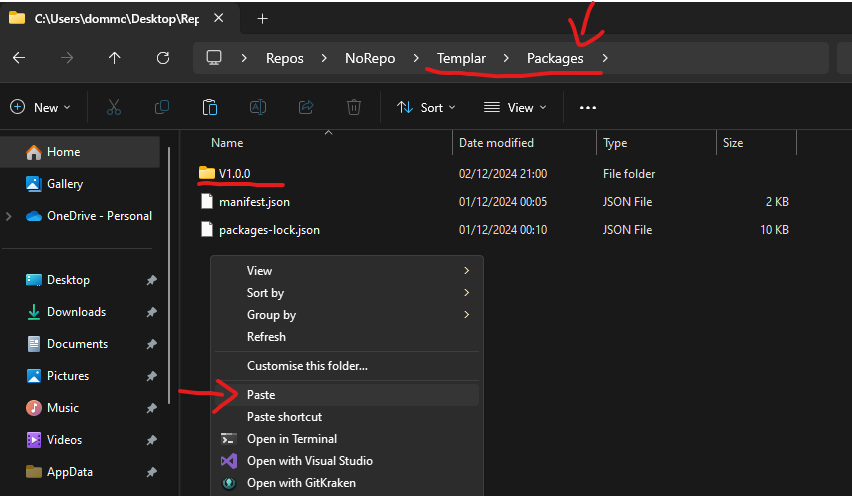
Once you have extracted the zip, please copy the folder with the package.json into your packages folder inside the unity project.



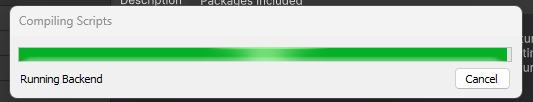
Copy the folder with all the package data to your project’s packages folder.

A screen shot of a computer

Description automatically generated



When you go back to the unity editor, it should recompile.



Once complete, you should see the package in the package manager.

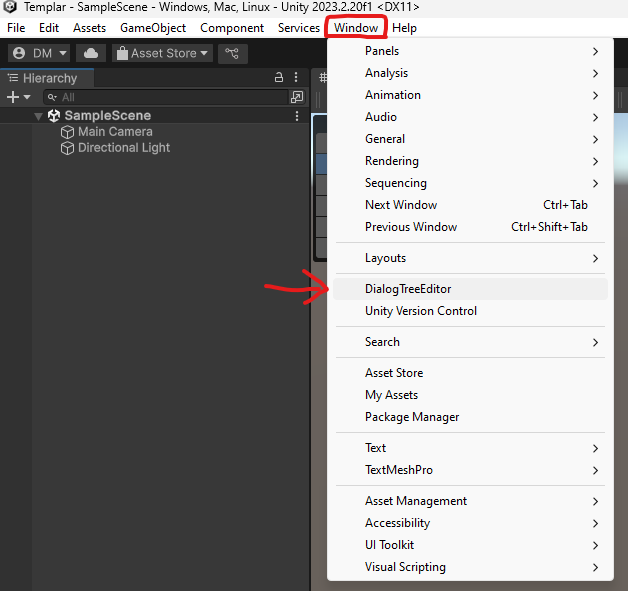


Once you have done that, you are ready to use the asset!

# Using the tool to create a dialog asset:

To start off, lest open the node-based editor.

Click Window > DialogTreeEditor to open the editor.



The editor window should look something like this.

A screenshot of a computer

Description automatically generated

Once open, you can start working on the dialog.

## Navigation:

You can use middle mouse button to move around the gird.

You can use scroll wheel to zoom in and out.

There is also drag selection as well.

A screenshot of a computer

Description automatically generated

## Creating a node:

Let’s start by creating a node for our dialog to start.

Right click anywhere inside the dialog tree editor and select Add Dialog Node.

A screenshot of a computer

Description automatically generated

## Breakdown of the node: (PUT UNIVERSE IMAGESSSS)

A screenshot of a computer

Description automatically generated

This is a dialog node; this allows you to display text from an option.

### Option display field:

The field in the top left corner is the text to display for the option, examples, yes, no, hello, bye, thank you. It’s up to you what the prompt for the player is.

These options are for what the player’s character says.

A screenshot of a computer

Description automatically generated

Example of the choices being displayed.

A screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generated

### Input and Output:

The input and output nodes allow you to create the dialog flow, you may want to have multiple responses or make the dialog feel more fluid and flexible.

A screenshot of a video game

Description automatically generated

To link a node to another node you just need to drag the output node from one dialog node to the other dialog node’s input.

A screenshot of a computer

Description automatically generated

A simple way is to link one to another. This is a linear dialog, no multiple choices, just one choice.

A screenshot of a computer

Description automatically generated

Example of single choice.

A screenshot of a computer

Description automatically generated

Now, if you want to have multiple choices, you can branch out to multiple nodes.

A screenshot of a computer

Description automatically generated

Here shows that the player can pick 3 options. They all will still end on one final node.

This is equivalent to semi-linear dialog.

Example of multi choice from multiple branches.

A screenshot of a phone

Description automatically generated

If you want to end on different dialog options, you can by ending on the branch.

A screenshot of a computer

Description automatically generated

Here the conversation will end on any of these three options.

// universe image.

### Dialog box:

Now, you want to add some actual dialog. To add some dialog, you just need to add text to the box that says, “DIALOG TEXT HERE”.

A screenshot of a computer

Description automatically generated

Click inside that box and you can start adding dialog.

A screenshot of a computer

Description automatically generated

## Start node:

The start node assigns a dialog node as the beginning piece of text.

A screenshot of a computer

Description automatically generated

It is required to be linked to the starting dialog node of choice. Otherwise, the program does not know the start node and cannot function properly. To link the start node to a dialog, you just need to drag the start output node to the dialog input.

A screenshot of a computer

Description automatically generated

# Tool bar:



## Saving:

The save button allows you to save the dialog tree as a physical file.



The graph name field is the name for the asset that is generated.

A screenshot of a computer

Description automatically generated

Clicking on the save button will prompt you to save the file in a location using your systems file system.

Important note, you must save inside the asset files otherwise it will reject you.

This creates a scriptable object that you can use to read in scripts.

A screenshot of a computer

Description automatically generated

### Note:

If you save the asset when it already exists and has scripts that references the asset, all references to the asset will be lost, so make sure to re add the references.

## Loading:

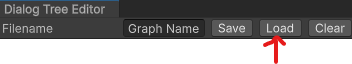
You can also load the asset to edit it.

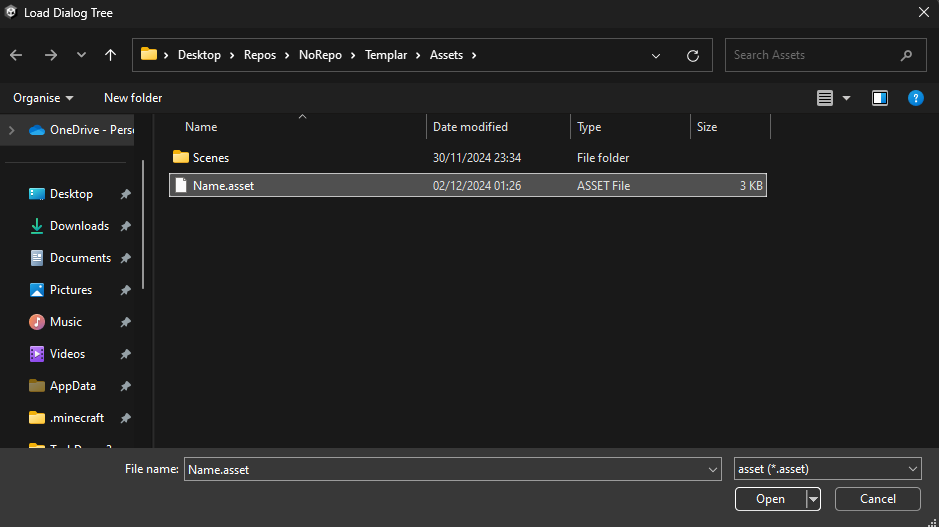
You just need to click the load button and select the dialog tree asset to open it.

(You can also double click the scriptable object or click open on the scriptable object)

A screenshot of a computer

Description automatically generated





A screenshot of a computer

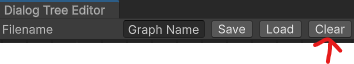
Description automatically generated

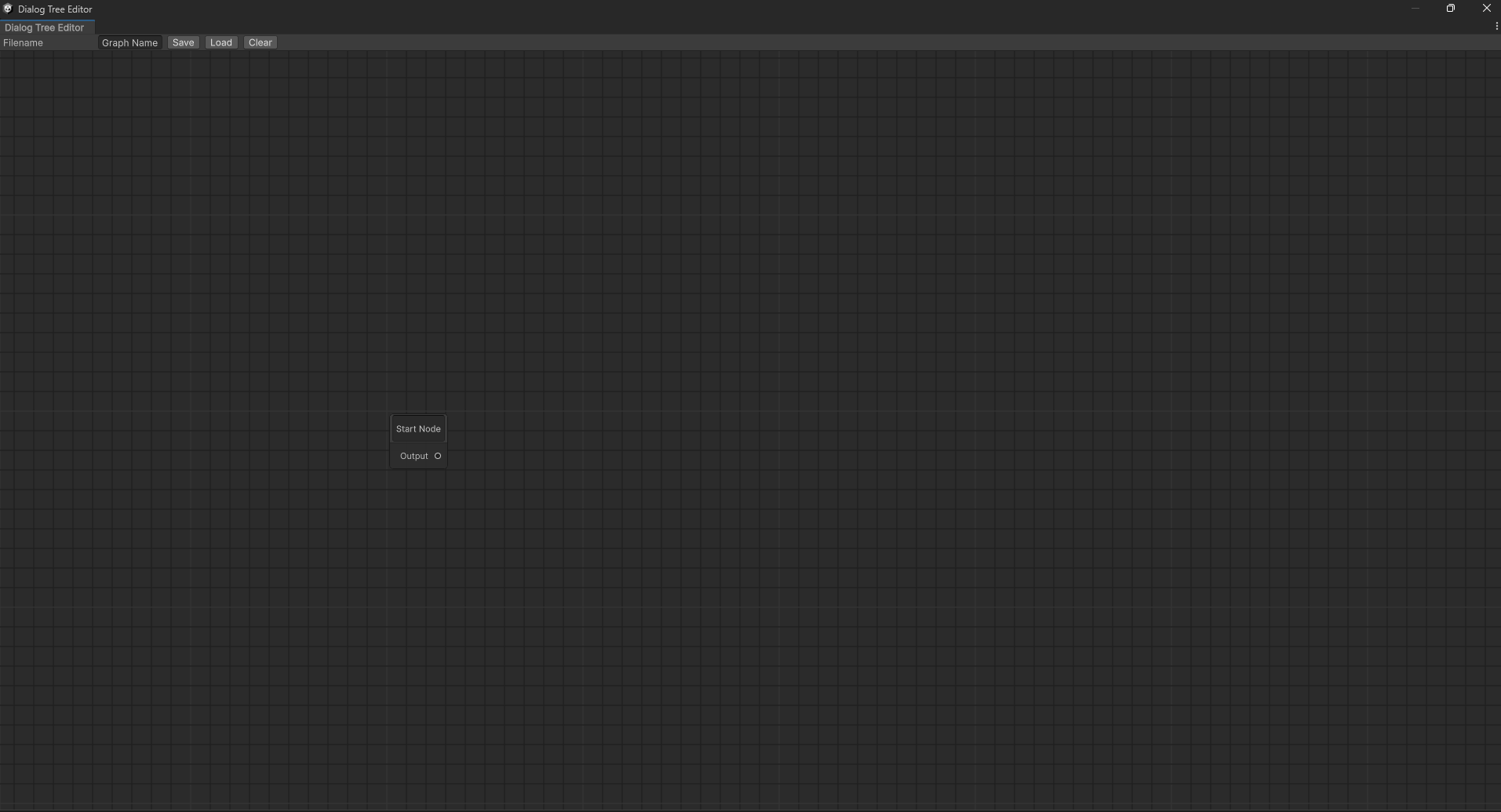
## Clear:

Well, it clears the view, this will remove any work you have not saved!

A screenshot of a computer

Description automatically generated





# Scriptable asset:

What is this thing? Well, this holds all the data from the dialog tree. You can use this asset for dialog.

Where is the original script located, If you added this package to the unity editor, it will be in Packages > Narrative Tool > Runtime > DialogTreeData

A screenshot of a computer

Description automatically generated

Everything located in the package is in the Narrative Tool folder in packages.

## SO Source Code:

<https://github.com/domibron/NarrativeToolUnity/blob/main/Assets/Runtime/DialogTreeData.cs>

This link goes directly to the SO in the GitHub repository.

## How to set up the example scene and run it:

I am going to use the example I made; you will also have this in your Packages > Narrative Tool > Samples > Example dialog

A screenshot of a computer

Description automatically generated

### DISCLAIMER:

The sample in the package will break, I will go through how to fix and use this sample. If you need a working one, please use the sample from the GitHub project as that will have all the references.

### Let’s fix the sample:

#### Button:

Double click the button in the prefabs folder in the sample. We want to open the prefab editor.

A screenshot of a computer

Description automatically generated

Import TMP.

A screenshot of a black box

Description automatically generated

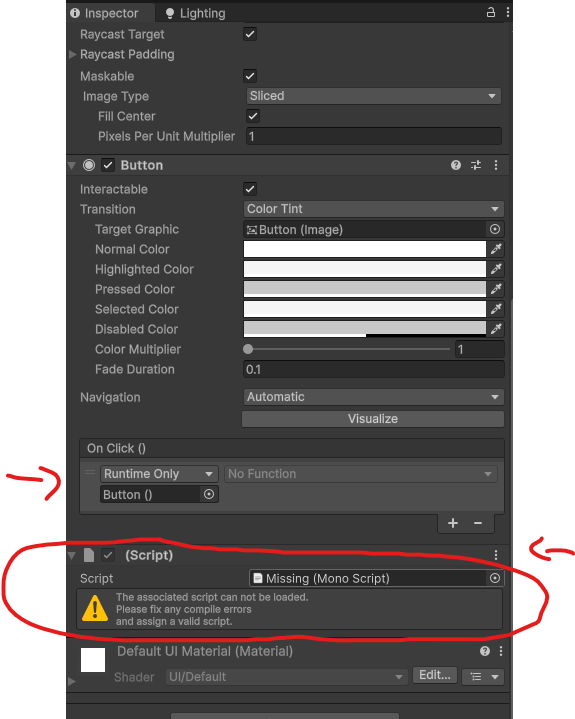
We do not need the extras.

You can the close the TMP window.

Click the button in the hierarchy. It will have a missing script. At the bottom of the inspector.

A screen shot of a computer

Description automatically generated



Drag the OptionItem script into that empty field. The Option item is located in the scripts folder in the sample.

A screenshot of a computer

Description automatically generated



Now, it should update to be a valid script.

A screenshot of a computer

Description automatically generated

Drag the attached Text (TMP) in the hierarchy to the scripts where it says Option Text.



A screenshot of a computer

Description automatically generated

Next is to drag the script to the button event field.

A screenshot of a computer

Description automatically generated

After that, we want to select the OnClicked function inside the script in the event field. To do this, click the drop-down, then click OptionItem, then OnClicked.

A screenshot of a computer

Description automatically generated

The button prefab is now done. Exit the prefab editor by clicking the right arrow next to the parent in the hierarchy.

A screenshot of a computer

Description automatically generated

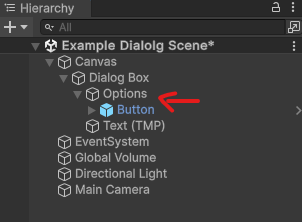
Select the red prefab in the scene hierarchy.



We want to replace the missing prefab with the one we fixed. Delete missing prefab in the scene then drag out the fixed prefab in the prefabs folder into the scene under the Options object.

A screenshot of a computer

Description automatically generated



Now we need to fix the canvas.

#### Fixing the canvas / dialog manager:

Select the canvas in the scene.

A screenshot of a computer menu

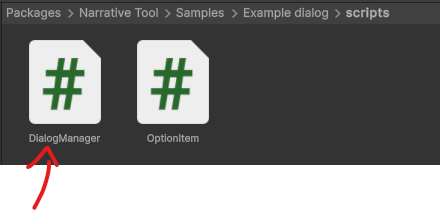
Description automatically generated

In the hierarchy look for the missing script component. Normally located at the bottom.

A screenshot of a computer error message

Description automatically generated

Drag the dialog manager into the missing script field.



A screenshot of a computer error message

Description automatically generated

It should now show field.

A screenshot of a computer program

Description automatically generated

Now we drag in the components.

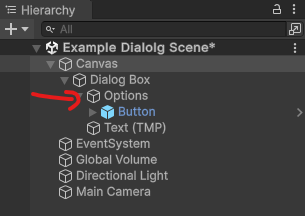
Drag in the example dialog scriptable object into the data field. The scriptable object is located in the project folder under Packages > Narrative Tool > Samples > Example dialog.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Drag the Options parent in the scene into the Options Parent field.

A screenshot of a computer program

Description automatically generated

Drag the fixed prefab in the prefab folder into the Options Prefab into the dialog manager attached to the canvas.

A screenshot of a computer

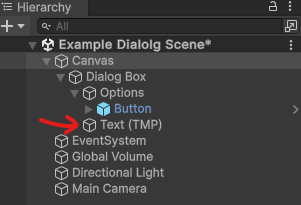
Description automatically generated

A screenshot of a computer

Description automatically generated

Now for the final field, the text box / Dialog Display.

Drag the Text (TMP) in the scene that is attached to the canvas as a child into the dialog manager.



A screenshot of a computer program

Description automatically generated

Now you should be all set.

If you want to test other dialog, change the Example dialog data out for another or edit it. This is up to you.

### Run the scene:

Click the play arrow that the top middle of the screen.



The text will change, and 2 options will appear.

A screenshot of a computer

Description automatically generated

Note, the done button will do nothing. This is meant to close the dialog / restart it, but this is not implemented. It just means that you can implement a close function or something else.

## Coding in the dialog data:

We need to make a public field so we can add the asset.

public DialogTreeData Data will suffice.



We also want to store the current dialog node.

private DialogNodeSaveData currentNode will do.



We can use Data.StartNode to get the starting dialog node.

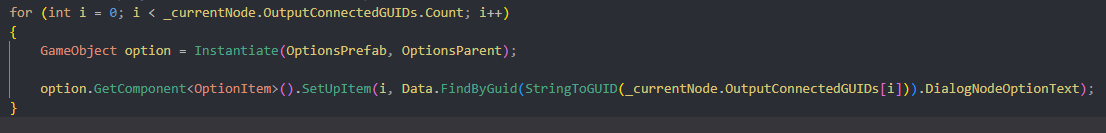
currentNode = Data.StartNode



We can use FindByGuid to get the node with that guid.

The currentNode will have all connected inputs and output nodes as guid stored in a list as strings.

You can use this to create option boxes for each output node attached to the current node.



Update dialog example.

A screenshot of a computer program

Description automatically generated

Because this dialog manager, it has a single ton. The generated buttons are assigned a int, they call SelectOption function and pass in their number to select the next node.

A screenshot of a computer program

Description automatically generated

Turning string into GUID example

A screen shot of a computer program

Description automatically generated

In case you need it.