Narramancer

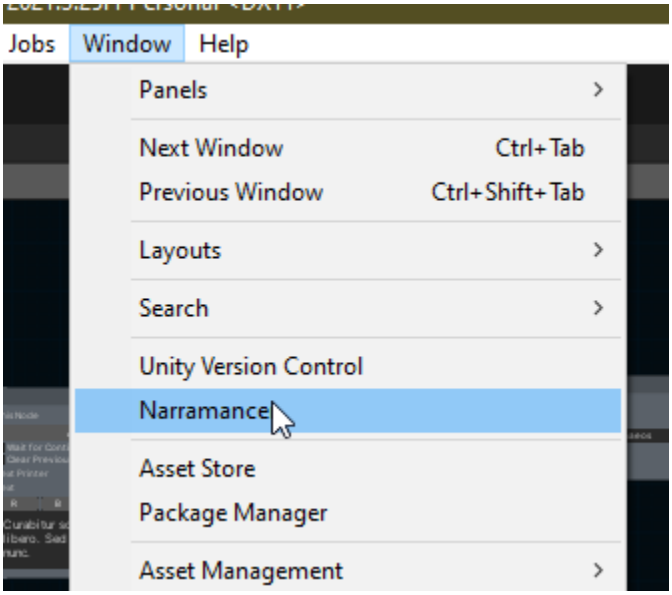
A node-based narrative system for Unity

1. Overview

Narramancer is a general solution narrative system with behavior trees and a built-in save system. It acts as a scene independent table of the people and things in your game, as well as what they are doing.

2. Getting Started

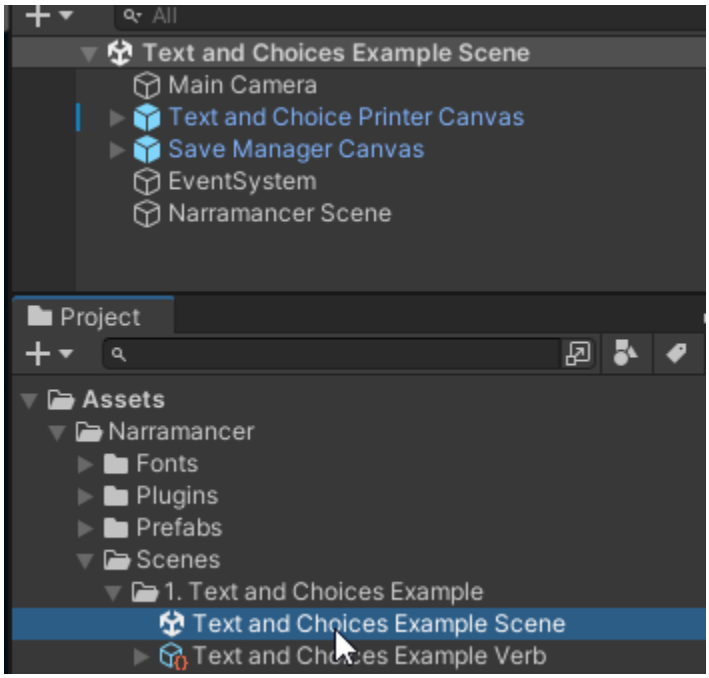
1. Import the Narramancer Unity package into your project
2. Go to Project Settings -> Player -> Other Settings and Change Api Compatibility Level to ".NET 4.x"
3. Open the Narramancer Window by choose 'Window -> Narramancer' from the menu bar
    - This will 'initialize' Narramancer and create the assets we need



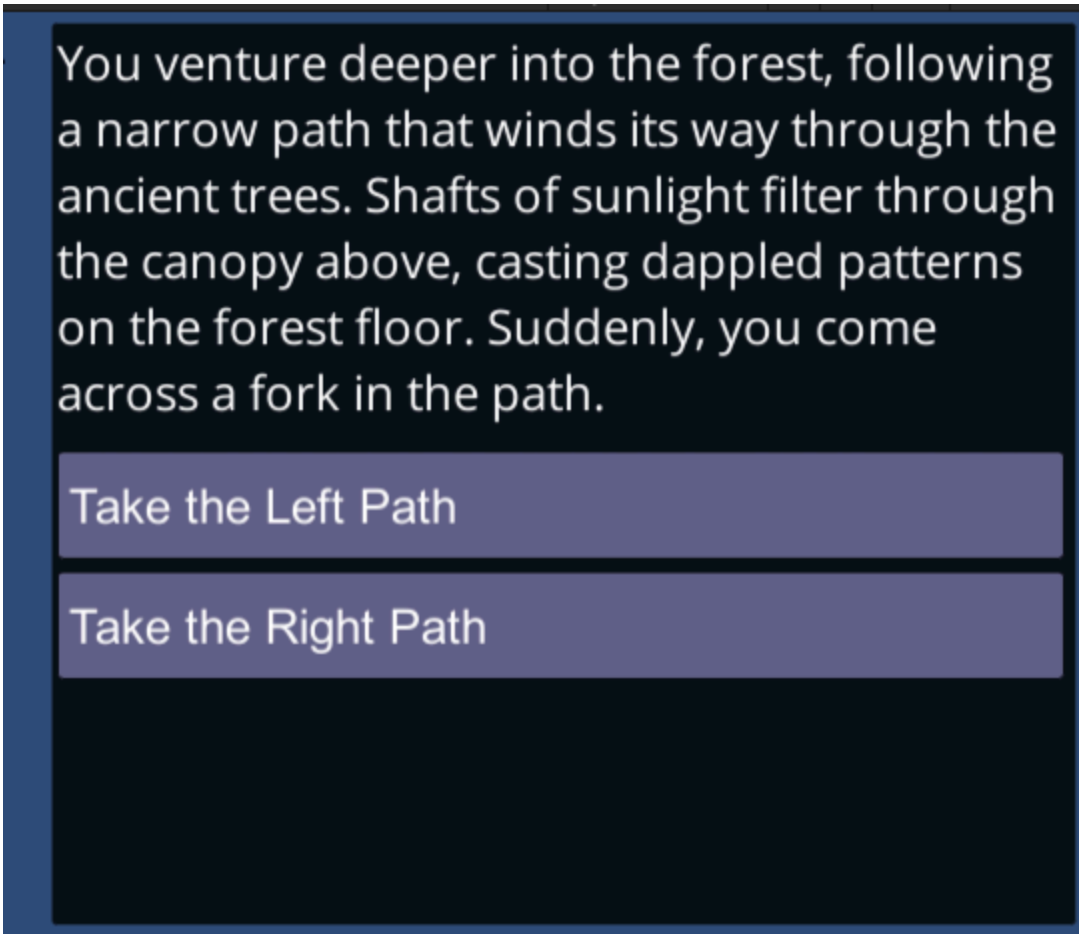
3. Narramancer Example Scenes

Narramancer includes a handful of example scenes demonstrating some basic implementations of the system .They are located in the folder Narramancer/Scenes.

Text and Choices Example



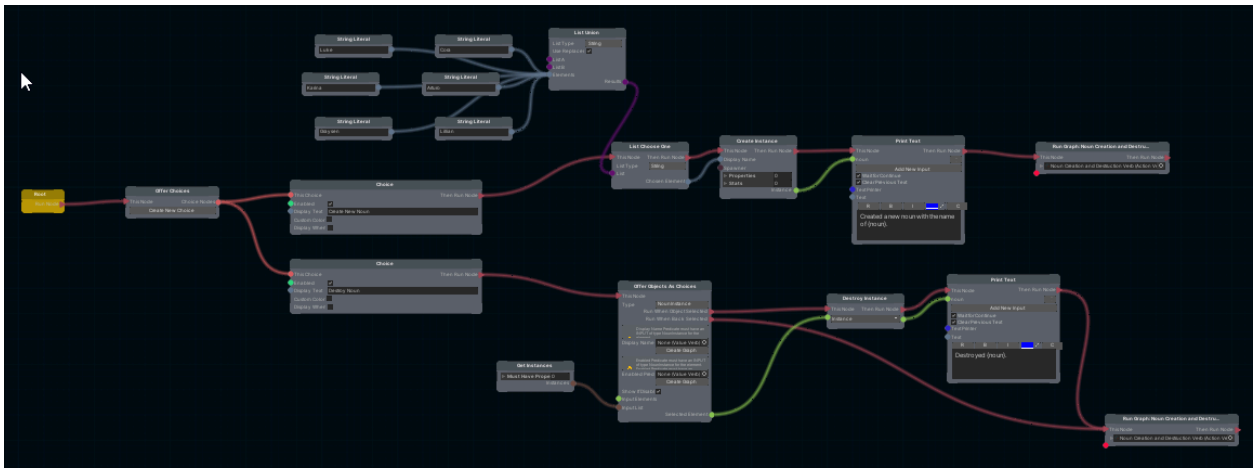
This example scene demonstrates simple text printing and choice selection. Upon entering Play Mode, the user is presented with a text dialog box which will progress when the user clicks on the screen. Eventually, the user will be presented with two choices, which will result in one of two text paths progressing.



The flow of this example scene is contained entirely within the ActionVerb asset named ‘Text and Choices Example Verb’. If you select the ‘Narramancer Scene’ object within the scene you will see the ActionVerb included in the list of ‘Run on Start Verbs’.

Other GameObjects of note in this scene are the ‘Save Manager Canvas’ and the ‘Text and Choice Printer Canvas’. Both of these are prefabs that can be extended, modified, and used in any scene. The Save Manager displays buttons and menus for saving and loading the player’s game. The Text and Choice Printer displays narrative text and choices for the player.

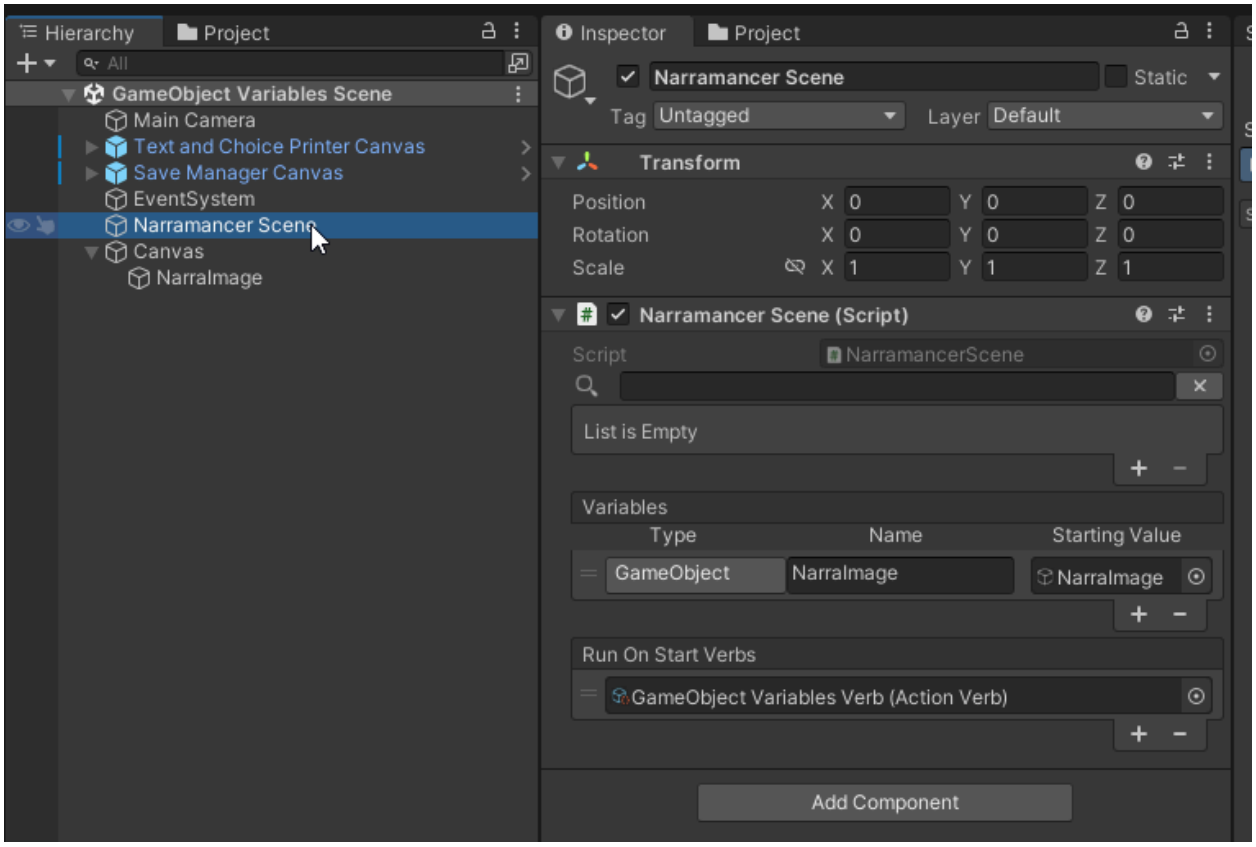
Noun Creation and Destruction



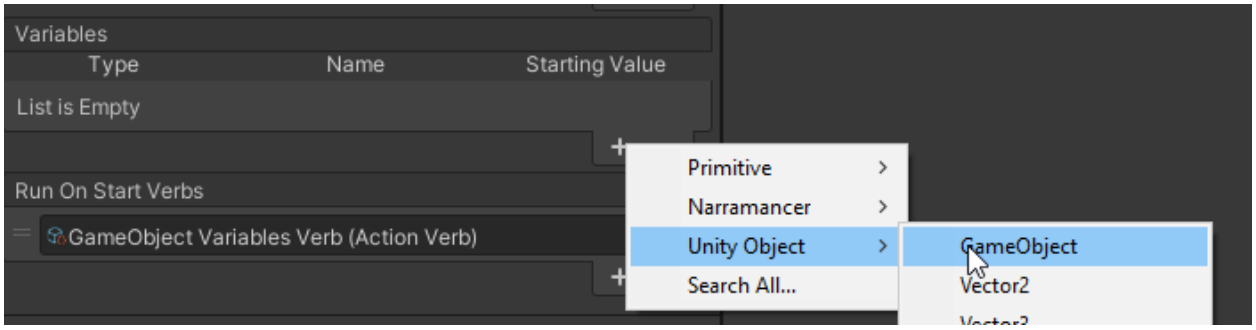
Running this example scene will present the user with two choices: Create a New Noun and Destroy a Noun. Selecting Create will cause a new noun instance to be created with a random name. Selecting Destroy will allow the user to choose from all existing nouns, afterwhich the selected noun will be destroyed. After either option the loop will repeat.

GameObject Variables

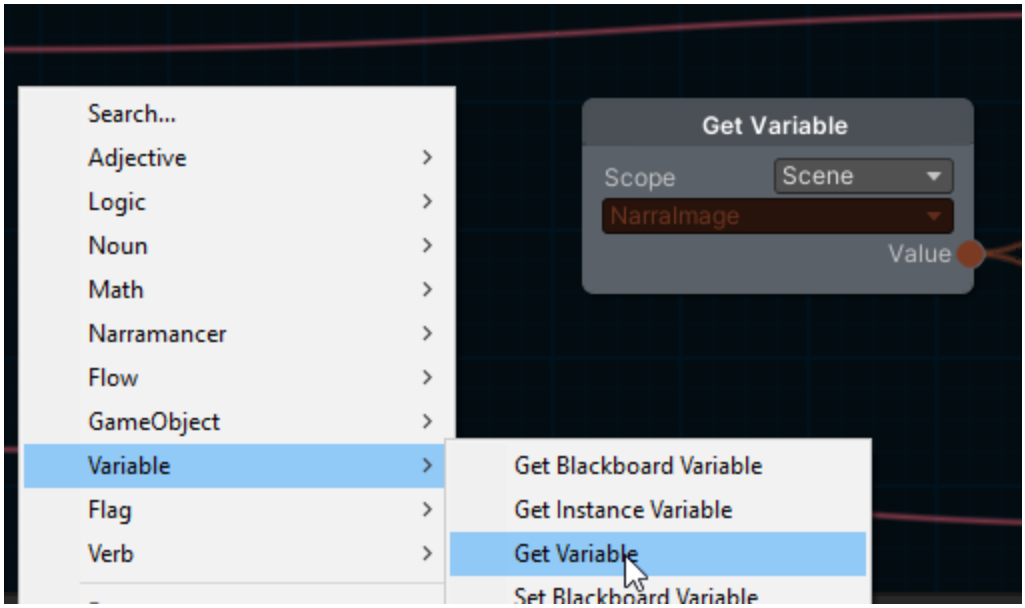
The purpose of this example scene is to demonstrate how to manipulate in-scene GameObjects from Narramancer Verbs. The best way (currently) is to assign the GameObject to a Narramancer Scene Variable.



Notice how the 'NarraImage' GameObject is assigned to one of the Narramancer Scene's variables. This can be done by either dragging the GameObject onto the Variables field (which will result in several entries being made, for each possible type you may want including the GameObject, Transform, and any components on the GameObject) OR by using the plus symbol to add an entry with the selected type.



You must then assign the Starting Value of the variable and give the variable a distinct name. You can then use the value of that variable from within a Verb using the GetVariable Node.



4. Terms

Nouns

The system assumes that the things in your game are ‘characters’ and provides a lot of convenience queries and operations to facilitate that, but at the same time allows for all kinds of things, including items, areas, ideas, etc. Because of this, the ‘things’ in Narramancer are referred to as Nouns.

Verbs

The most powerful feature of Narramancer is the Verbs, which is a way of representing what the things in your game are doing. Verbs come in two types: verbs that ‘get’ information called ValueVerbs, and verbs that perform actions over time (possibly a fraction of a second, possibly minutes or hours) which are called ActionVerbs. Both kinds of verbs are graphs built up of nodes. Verbs can also have any number of inputs and outputs. You may have one verb for the entire game that handles the logic from one mode to another, or you may have a verb attached to each of the characters that controls only what they are doing, or some kind of combination.

Verbs can be thought of as functions in C# or any programming language, with input parameters and any number of output result values. ActionVerbs are similar to methods that are ‘async’ and ValueVerbs are executed instantaneously like typical, non-async methods.
Verbs can be built that use other Verbs, executing them over time or using them to process or retrieve information. Note that ValueVerbs cannot use ActionVerbs, but ActionVerbs can use ValueVerbs.

Adjectives

Narramancer treats nouns similar to entities in Entity Component Systems, where a noun is simply a named container. Interesting data can then be added to a noun by simply attaching 'components' to it. In Narramancer, these kinds of components are referred to as 'Adjectives' and they come in three flavors: Properties, Stats, and Relationships.

Properties

Properties can be added to or removed from a noun over the course of the game, and can be used to group nouns together, or flag a noun as having a certain quality or being treated in a certain way.

Stats

Stats are number values that can be attached to a noun, then modified over the course of the game. Use stats to represent a noun's health, mana, strength, or even as a running count of how many battles this noun has been in.

Relationships

Relationships are unidirectional connections between two nouns. They can be added or removed over the course of the game, and are useful ways to represent things like how characters are related (eg: siblings), what has happened in the history between two characters, or how two areas are connected to each other.

Variables / Blackboards

In addition to adjectives, nouns each have a general purpose 'blackboard', which is a keyed table of values that can be read and manipulated at runtime. This is useful for holding numbers, strings, images, or any other kind of data that a noun might need either from the start or added during the game.

In fact, there are various blackboards that can be used for storing values, including on a Verb or on the global state of the game.

Save and Load

Saving in Narramancer takes the nouns plus any verbs and serializes them down into a save file, that can then be loaded at a later time. Verbs are built in such a way that even if it is in the middle of running an action, it remembers where it was and what it was doing, and can pick up right where it left off when re-loaded.

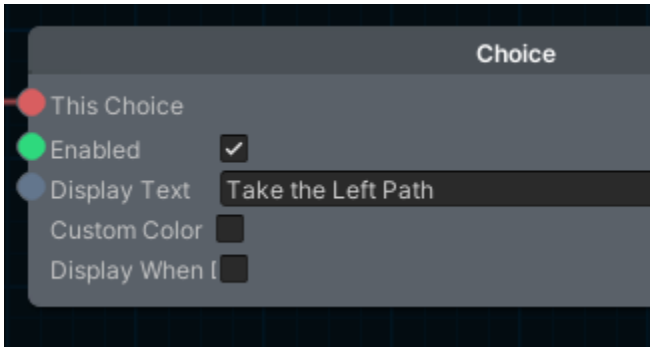
Instances

An important distinction to make is between the various assets and their runtime 'instances'. During Play Mode, Narramancer will create an 'instance' of every Noun and Adjective, which have all of the predefined aspects of the asset, but which allows the instance to be manipulated without affecting the starting asset. This also allows for one asset to spawn multiple instances.

Nodes

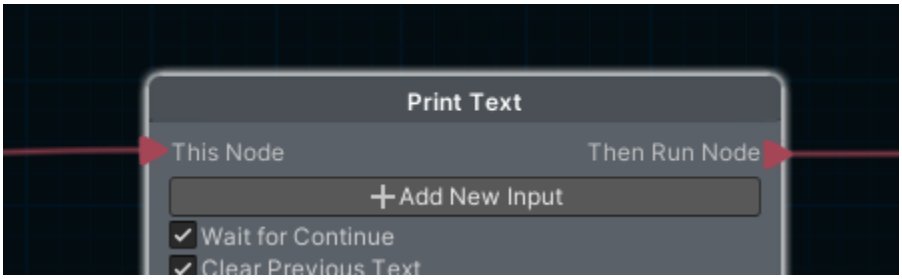
Verbs are made up of various nodes. Each node has a specific function. Nodes have one or more ports, and ports can either be an input or an output. An output on one node can be connected to an input on another node, causing the second node to use the output value as its input value. Input ports (usually) can only be connected to one output port at a time, but output ports can (usually) be connected to multiple input ports.

Ports always have a C# type behind them. That is, they are either numbers, or strings, or NounInstances, or Lists, etc. Ports can only be connected to ports that are the same type (or at least compatible). Input ports that are primitive values, such as ints, floats, or strings, usually allow for a literal value to be assigned to the input. For example, in the following image, the 'Enabled' input port is a bool and can be assigned to 'true' or 'false' simply by toggling it. The same can be said about the 'Display Text' port: the display text can be assigned either by connecting something to the port OR by typing text into the input text field.



RunnableNodes

ActionVerbs have access to a category of nodes referred to as RunnableNodes. Nodes that are ‘runnable’ have two special ports: one input port referred to as ‘This Node’ that causes the node to ‘run’ and one output port that, upon completing its own run, triggers the next runnable node to run. These special ports have triangle/arrow shapes to help differentiate them.



While in Play Mode, the NodeEditor will highlight RunnableNodes that are currently running or have run recently.

ActionVerbs must have exactly one RootNode, which connects to one RunnableNode and tells the verb which node to run first when the verb is run.

Narramancer Scene

A Narramancer Scene is a component that can be added to a scene that tells the Narramancer system key aspects about the given scene. Only one Narramancer Scene component at most should exist in a given scene.

The three main parts of the Narramancer Scene are a list of Nouns, a list of Variables, and a list of ActionVerbs.

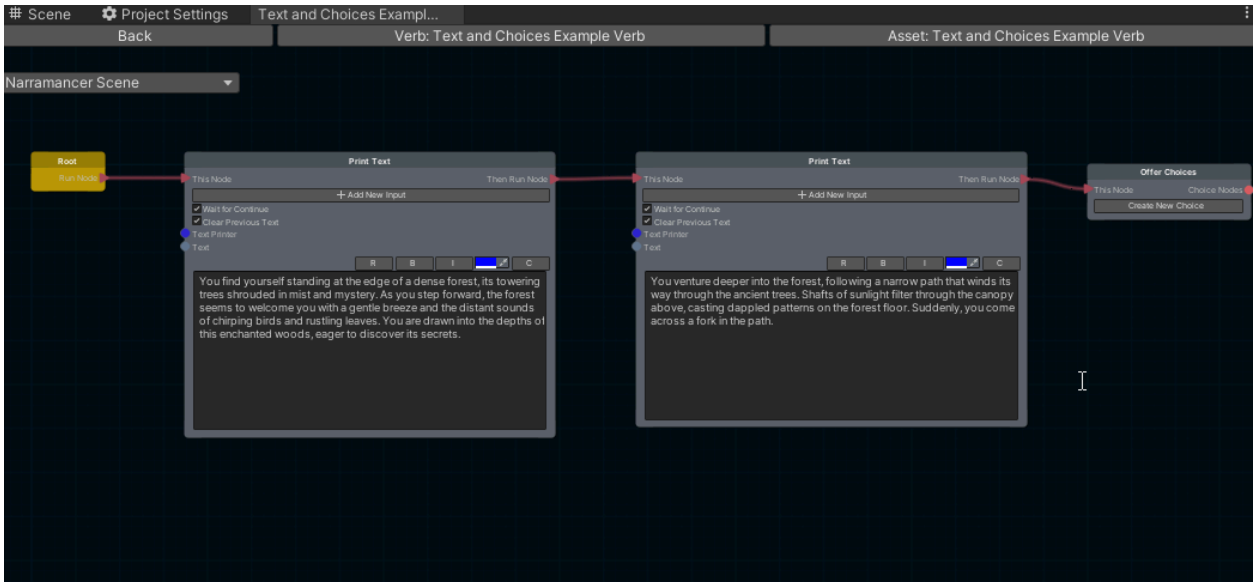
When the scene starts:

1. The given Nouns will be created/initialized.
2. The given Variables will be created and assigned to their starting value.
3. The given ActionVerbs will begin running.

Narramancer Scenes are useful for running ActionVerbs that depend on GameObjects from a specific scene by assigning the given GameObject(s) to corresponding Variables.

5. Verb Editor

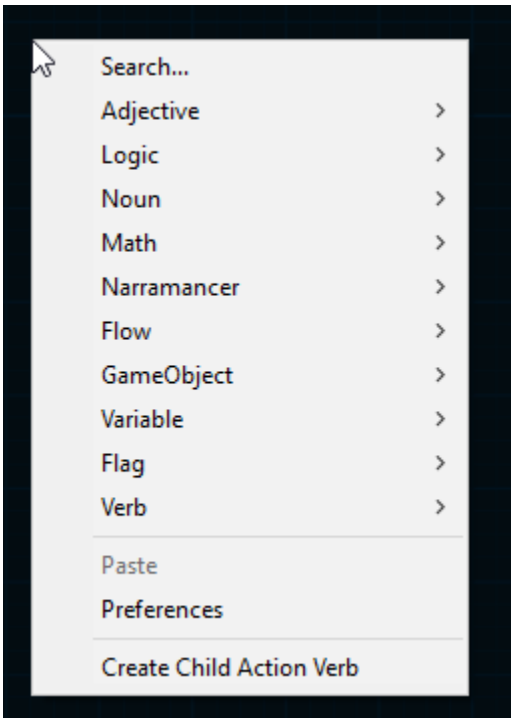
The most useful tool of Narramancer is the NodeGraph editor used to manipulate verbs. To open the NodeGraph editor window, double click on any Verb asset (ValueVerb or ActionVerb) OR select a Verb asset and click on ‘Edit Graph’ in the asset’s inspector.



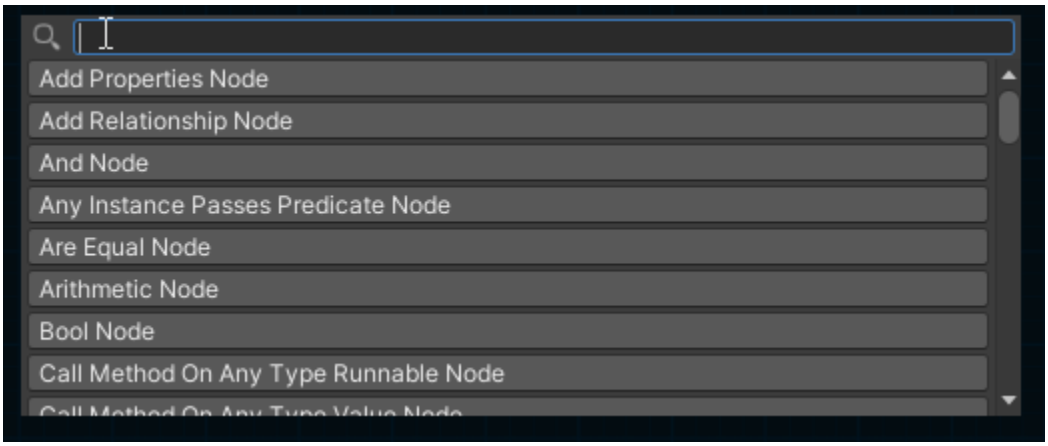
NodeEditor Window Basic Controls

- Mouse Scroll -> Zoom In and Zoom Out
- Press Middle Mouse Button and Drag -> Pan Visible Area
- Press F Key -> Jump to the ‘origin’ of the visible area
- Click Left Mouse Button -> select a Node
- Press Left Mouse Button on an empty area and Drag -> start a Selection Box, selecting all encompassed Nodes.
- Press Left Mouse Button on a node and Drag -> drag or translate the selected node or nodes.
- Press Left Mouse Button on a node’s output port and Drag, releasing on another node’s input port -> create a connection between the two nodes by the given ports.
- Click Right Mouse Button -> bring up the Node Context Menu

Node Context Menu



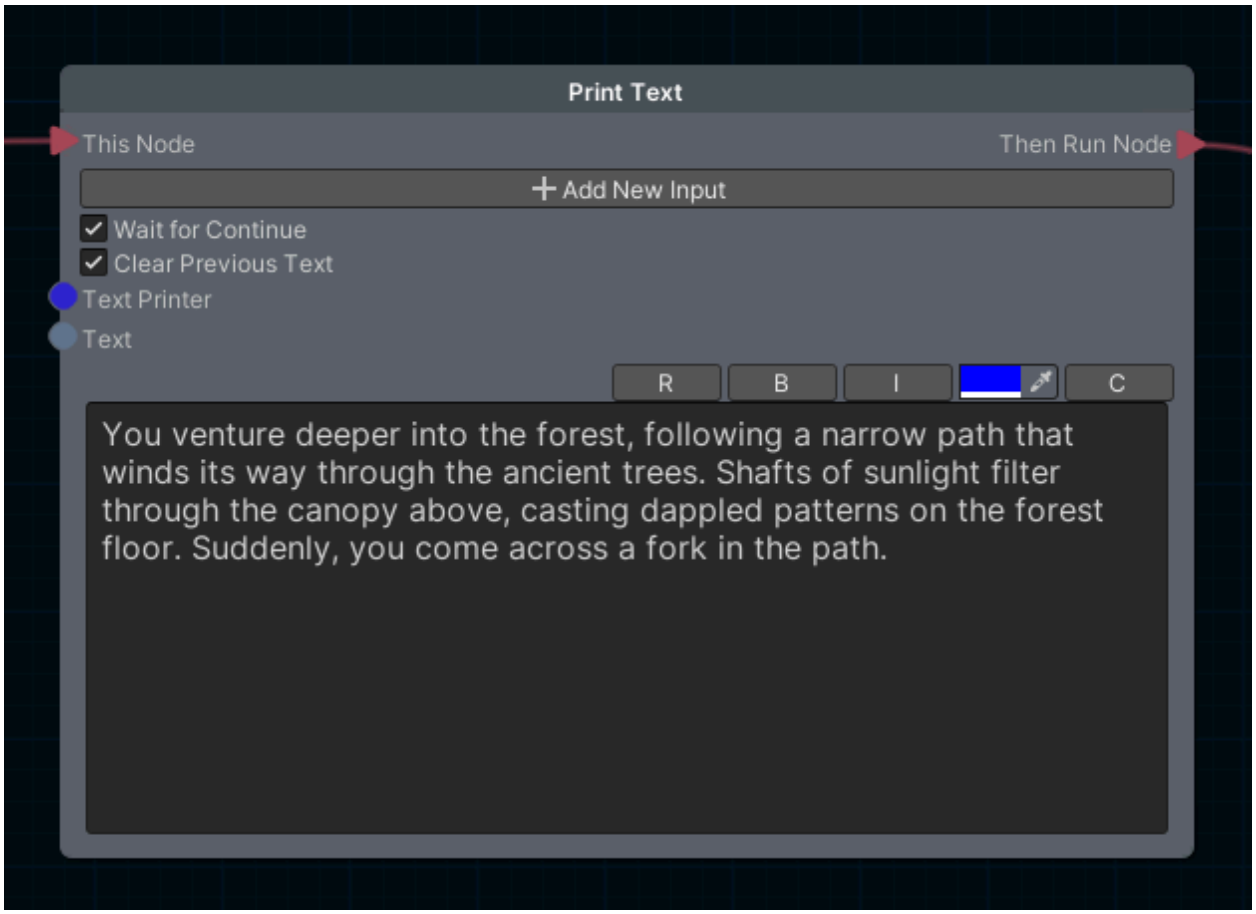
The Node Context Menu allows you to create and add new nodes to the given graph at the given position. The various possible nodes are organized by function. To find a given node quickly, choose the 'Search…' option and use the Node Search Dialog to find the node by its name by simply typing it.



Complex logic and functionality can be achieved by connecting various nodes.

Common Nodes

Print Text Node



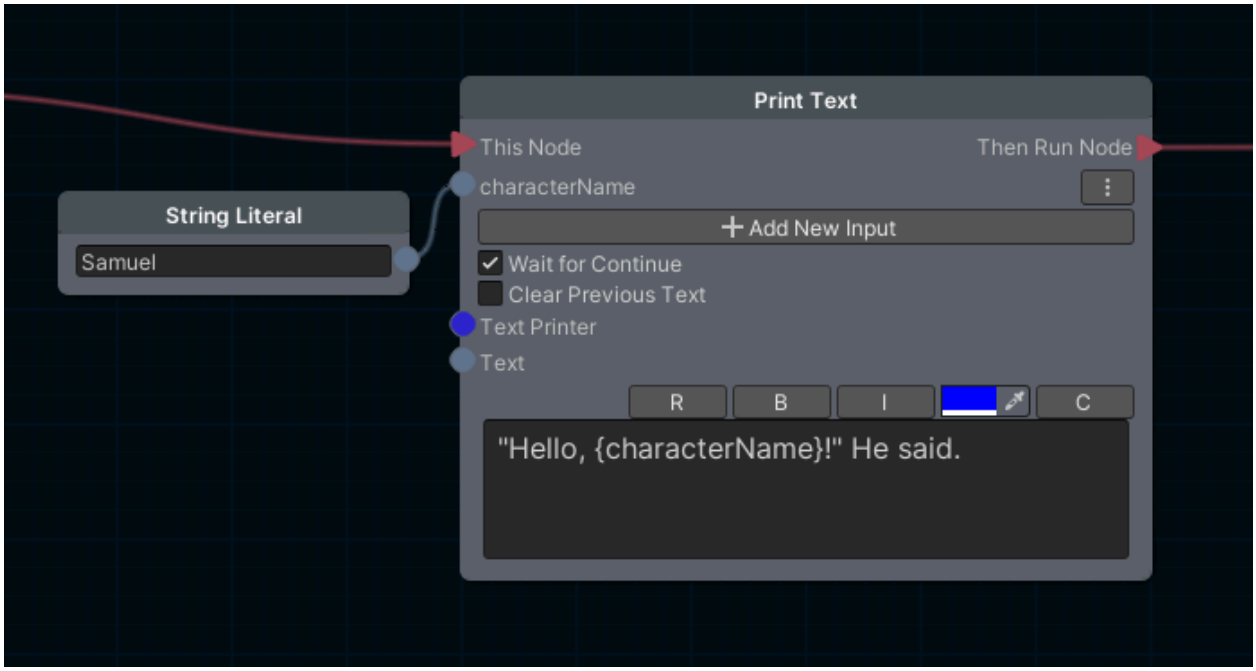
The Print Text Node is a RunnableNode that will display narrative text to the player.The text can either be typed into the text editor field OR given as an input by connecting the ‘text’ input port. A Text Printer can also be given, which will determine the GameObject that receives the text, but if none is set then the first one found in the scene will be used.

The text editor field supports rich text tags, and provides convenience buttons for added tags around selected text. The ‘R’ button will enable or disable the rich text being displayed in the editor. The ‘B’ button will add (or remove) bold tags around the selected text. The ‘I’ button will add (or remove) italic tags around the selected text. The ‘C’ button will add (or remove) color tags for the color currently selected by the color picker around the selected text.

The Print Text Node can also be resized by selecting the node and dragging the bottom right corner.

The Print Text Node also allows for dynamic terms within the text using inputs. Use the ‘Add New Input’ button to create an input for any given type. Give the input a name, then use that name wrapped

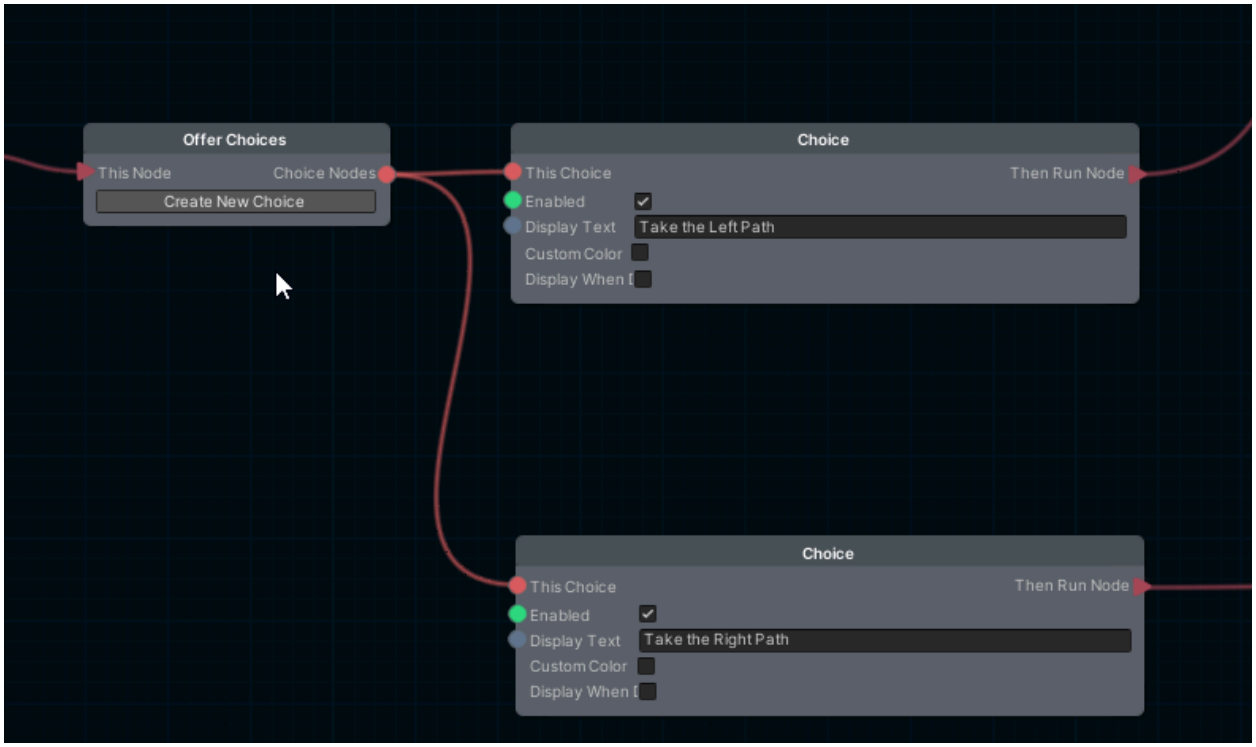
in curly brackets within the text to have that term replaced with the string representation of the input value.



The example given in the above figure will result in the text output to be

“Hello, Samuel!” He said.

Offer Choices Node + Choice Node



The Offer Choices Node and its companion, the Choice Node, allow you to display selectable choices to the player that will affect which branch runs next. The Offer Choices Node is a runnable node that will gather all Choice Nodes connected to it and display them as buttons to the first Choice Printer GameObject it finds in the scene.

The Choice Node has various options that affect how it is displayed. If 'Enabled' is set to false then the option will not be displayed at all UNLESS 'Display When Disabled' is set to true, in which case it will be displayed but in a disabled style.

The RunnableNode connected to the Choice Node that is selected by the player is the one that will then be run by the system.

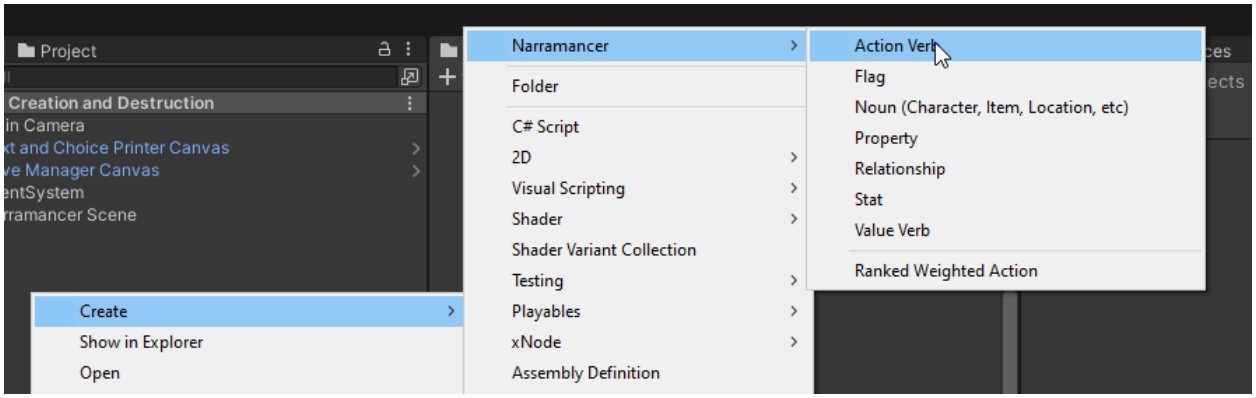
The Offer Choices Node can have any number of Choice Nodes connected to it.

6. How to…

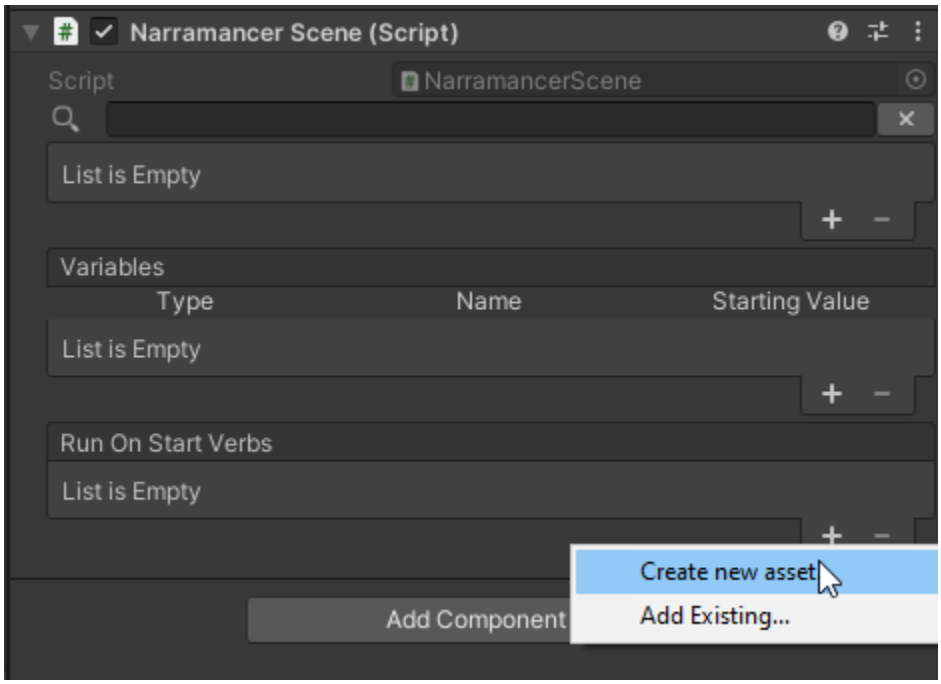
Create ActionVerbs

There are two main ways to create new Verbs:

1. Right-click anywhere in the project and use the Asset Creation menu to create a new verb asset, either a ValueVerb OR and ActionVerb from under the Narramancer menu



2. Create and add an ActionVerb to a Narramancer Scene component by using the plus button and selecting ‘Create new asset’:



Edit or Modify Verbs

Please refer to the [Verb Editor Section].

Run ActionVerbs

Create Adjectives

Create Nouns

Access GameObjects in the Scene from within a Verb

Please refer to the GameObject Variable Example Scene Section.

Create Custom Nodes