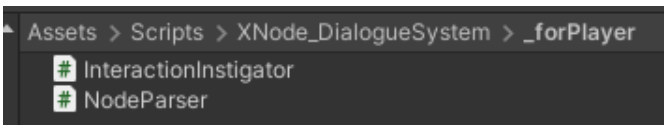
xNode Dialogue System

Jovan Catampongan | s3715622

| Setting it up

xNode Dialogue System allows you to add dialogue in your game with quick ease. No coding needed, all you have to do is to add the scripts and connect the nodes. Unless you want to make your own nodes and adapt it to your game.

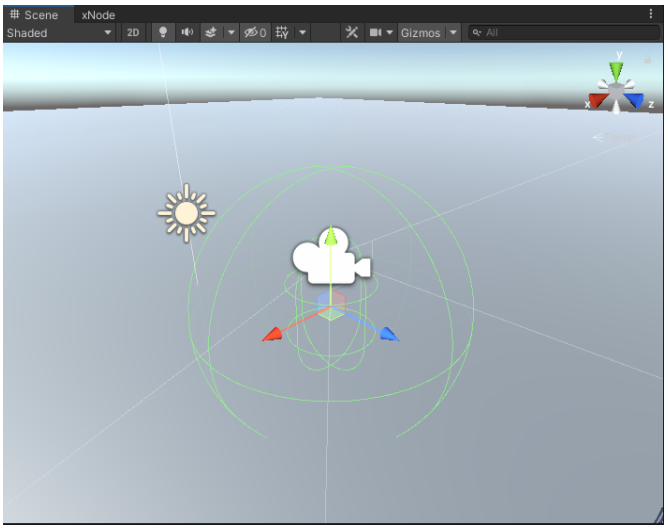
1. Your PlayerController GameObject must have the following scripts:
- NodeParser.cs & InteractionInstigator.

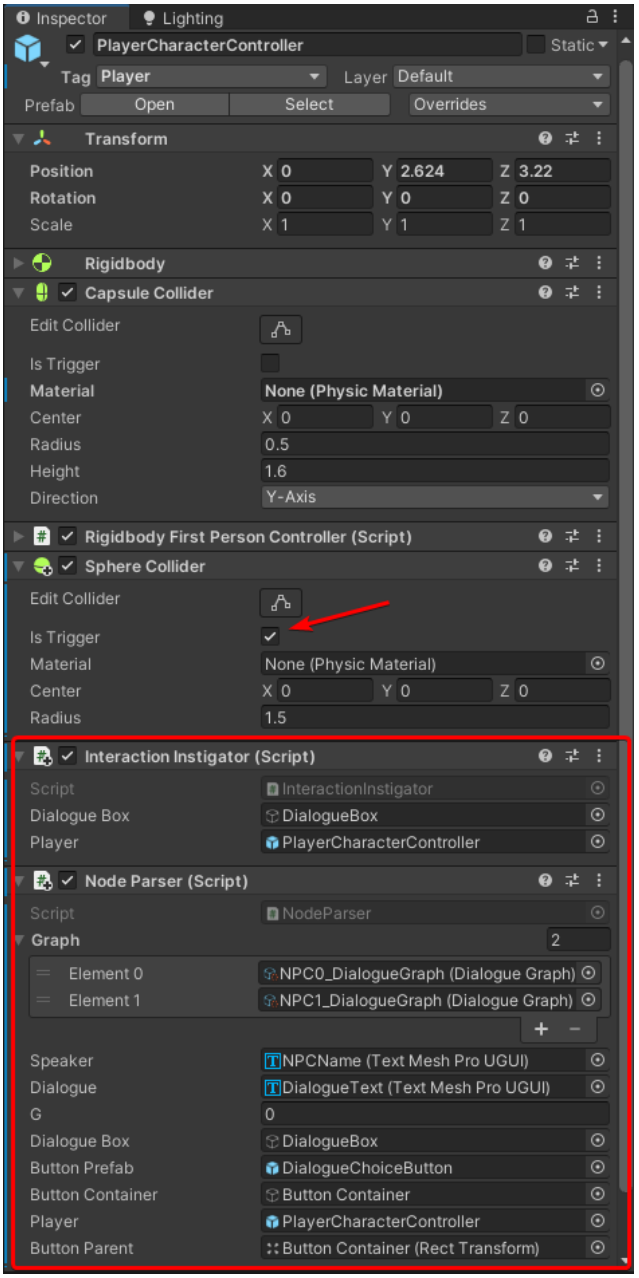


These scripts are in *Assets\Scripts\XNode_DialogueSystem_forPlayer*

Make sure you set your player tag is **Player**

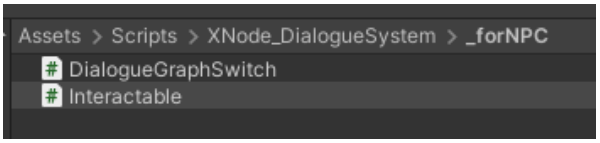
2. Add a Sphere Collider with 'Is Trigger' ticked on the PlayerController (as shown on the red arrow).



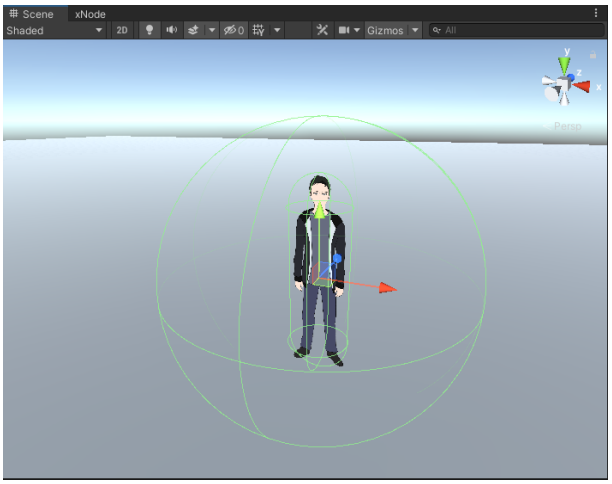


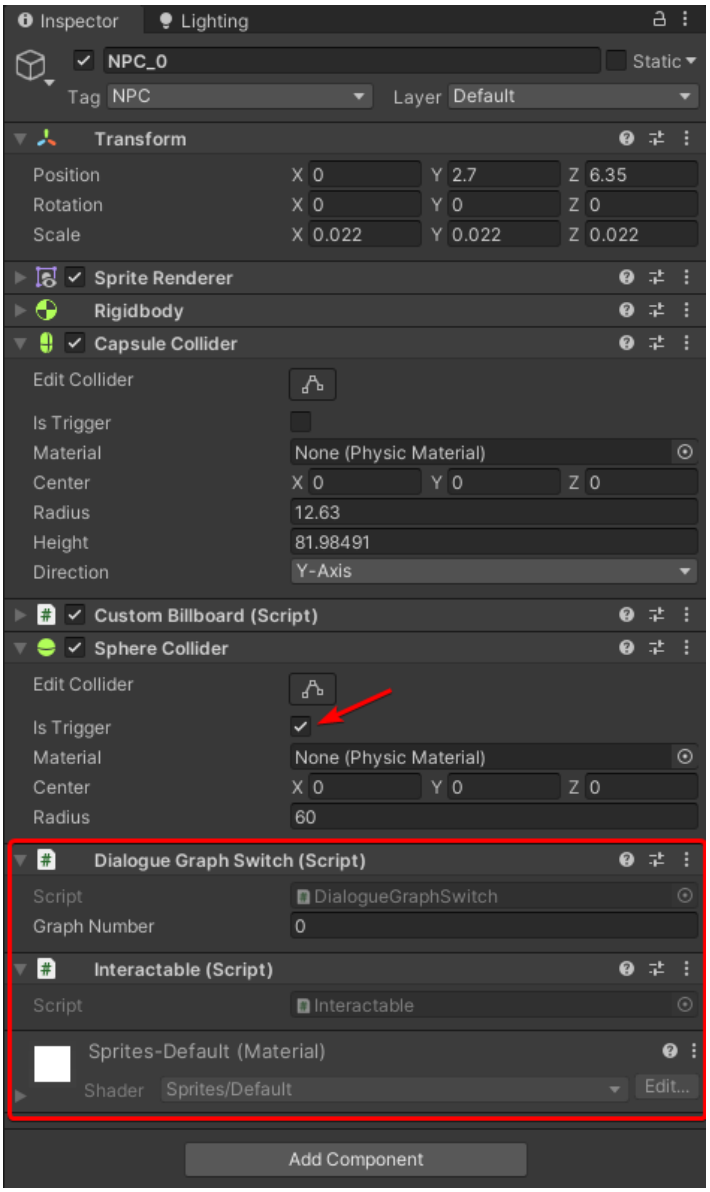
3. Your NPC GameObject must have the following scripts:
- DialogueGraphSwitch.cs & Interactable.cs

These scripts are in
Assets\Scripts\XNode_DialogueSystem_forNPC

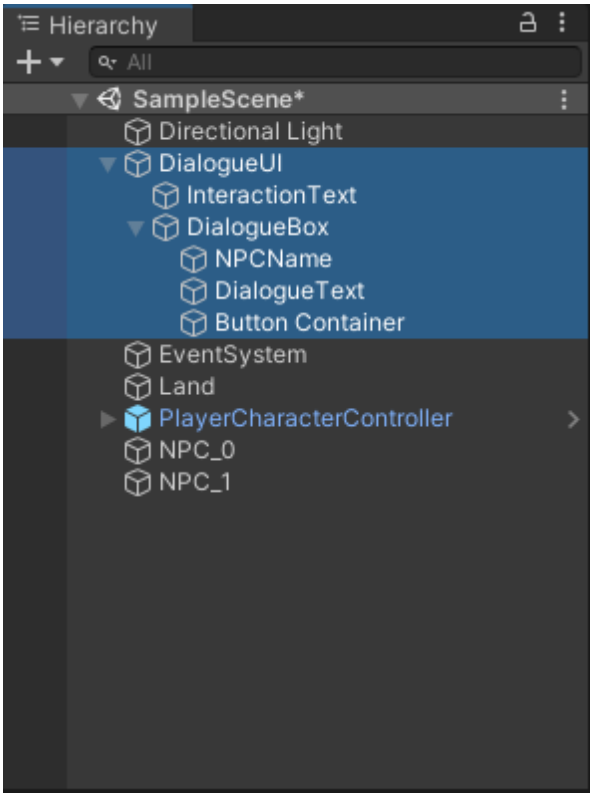


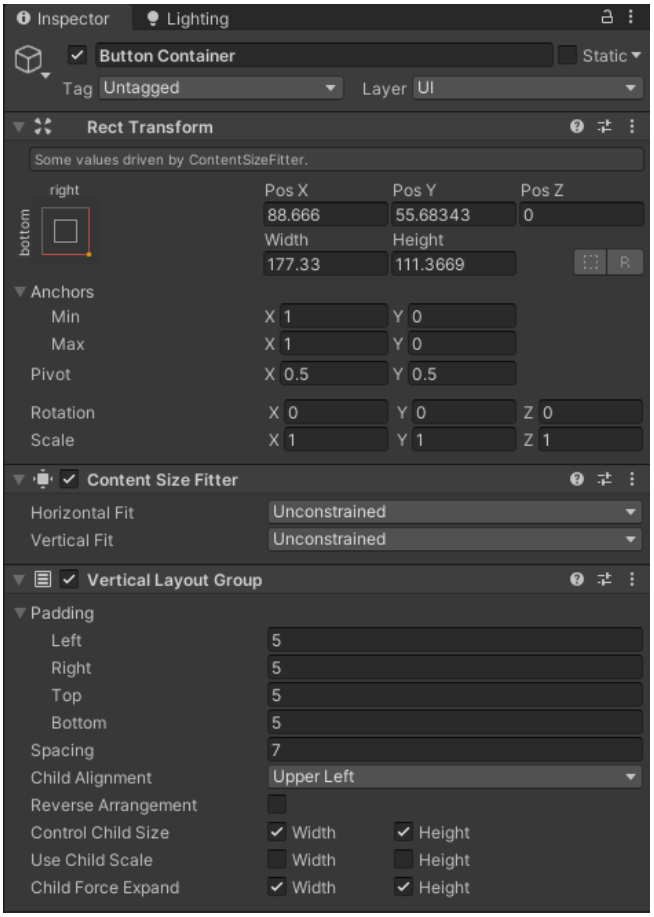
4. Add a Sphere Collider with 'Is Trigger' ticked on the NPC (as shown on the red arrow). Also, set the 'Graph Number' in sync to what you've added in the Graph array at NodeParser. For now, set your first NPC to 0





5. To create the DialogueUI, right click on the hierarchy and go to *UI → Canvas*
6. To create the DialogueBox, right click on the hierarchy and go to *UI → Panel*
7. To create the InteractionText, NPCName and DialogueText. right click on the hierarchy and go to *UI → Text - TextMeshPro* **(You must download TextMeshPro if you haven't already).** Adjust the UI accordingly.



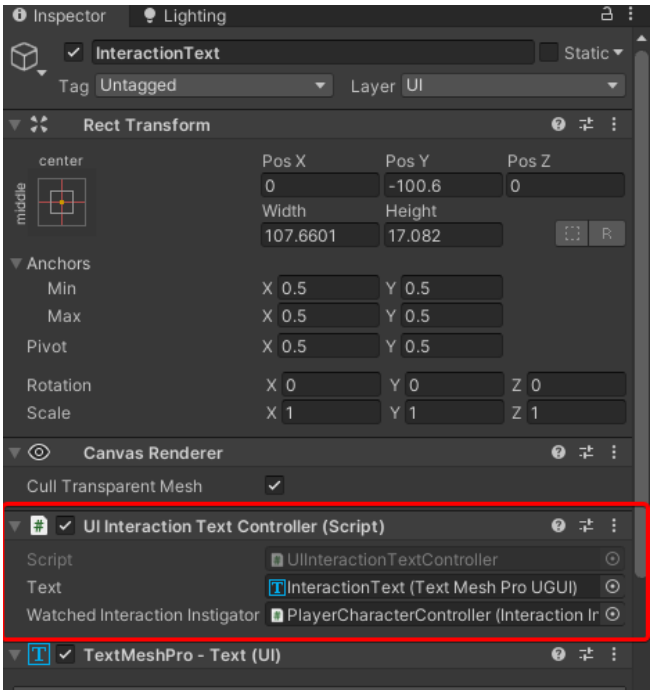


8. The Button Container consists the following components: Rect Transform, Content Size Fitter and Vertical Layout Group/

9. Add UIInteractionTextController.cs to InteractionText. Accordingly, add the GameObjects into the script as shown on the right.

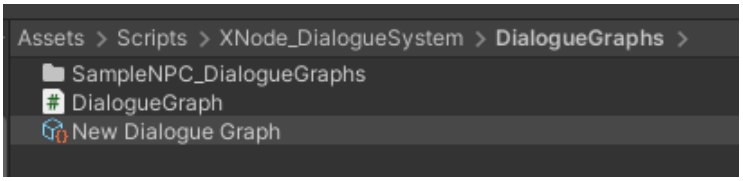
10. Add in the GameObjects into every attached script by following the screenshots. You could look at the SampleScene and compare. Or if possible, copy and paste the components into your Scene.

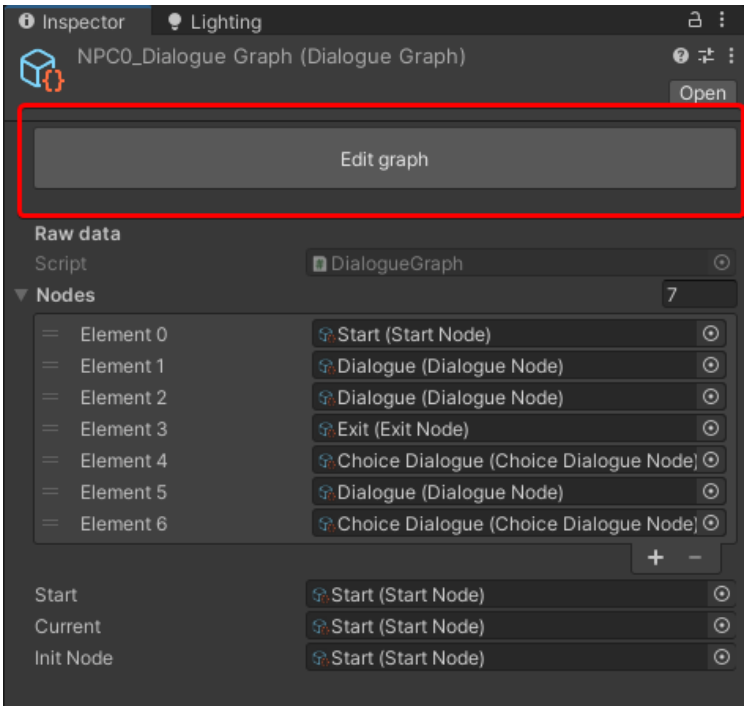
11. For the ButtonPrefab, just copy and paste it into your project folder.



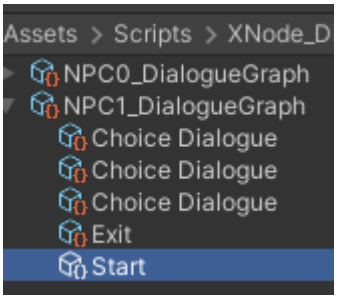
12. To get the Dialogue System working, you must create a DialogueGraph by right click in the Project Window and go to *Create → DialogueGraph*

13. The DialogueGraph is where you edit the writing of an NPC and create DialogueNodes and ChoiceDialogueNodes.

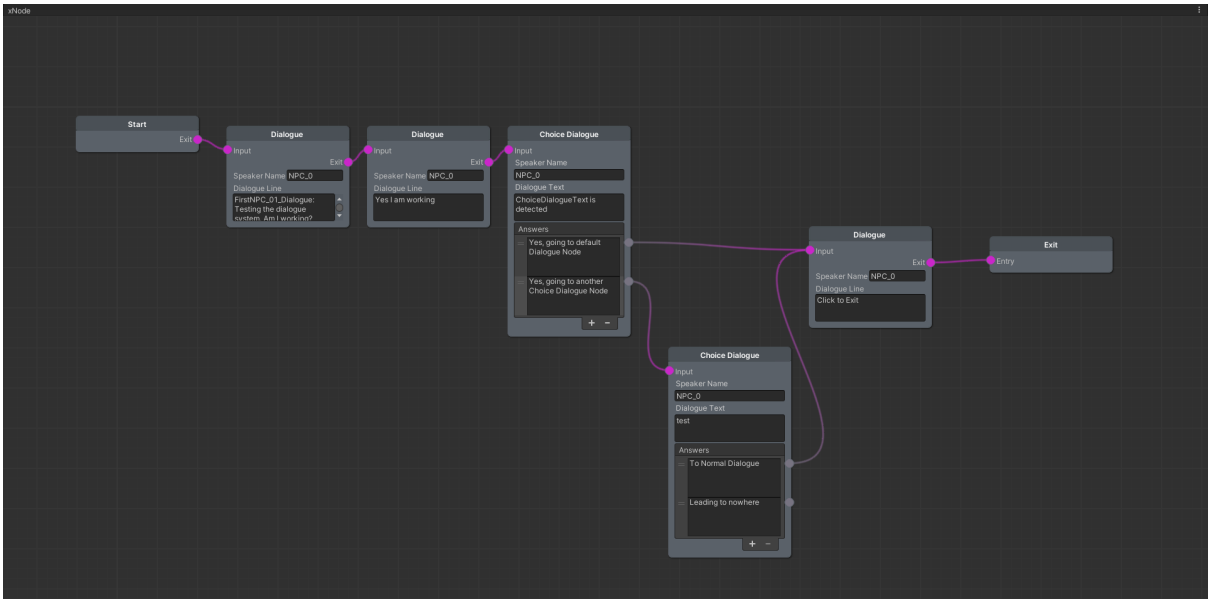




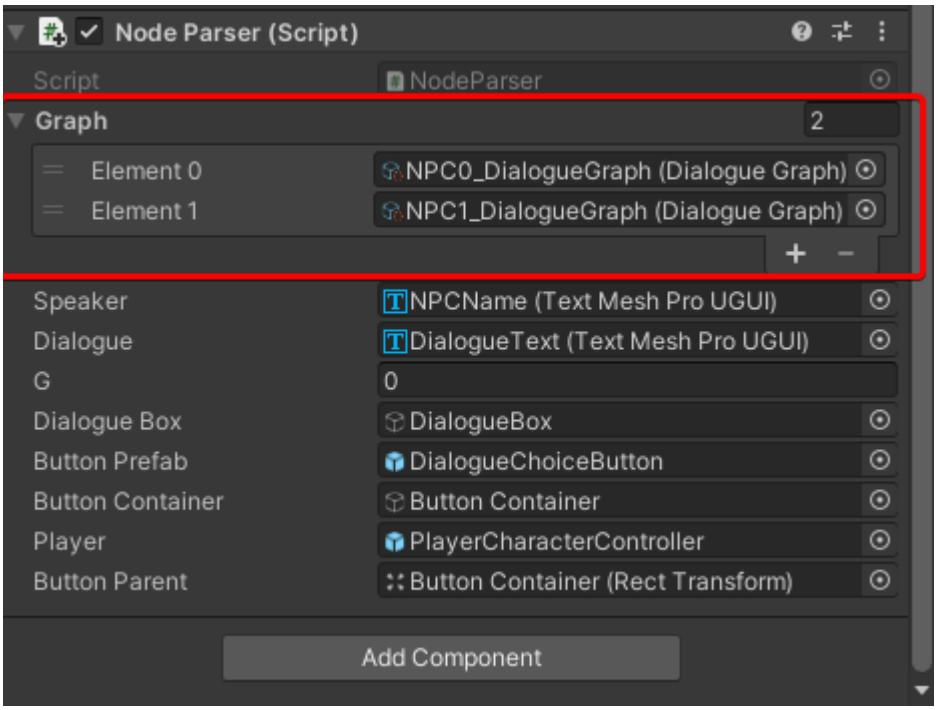
14. Click on your new DialogueGraph then click 'Edit Graph.' After you create your first nodes, make sure you select the graph's Start node and set it in 'Start, Current and Init Node' through drag and drop.



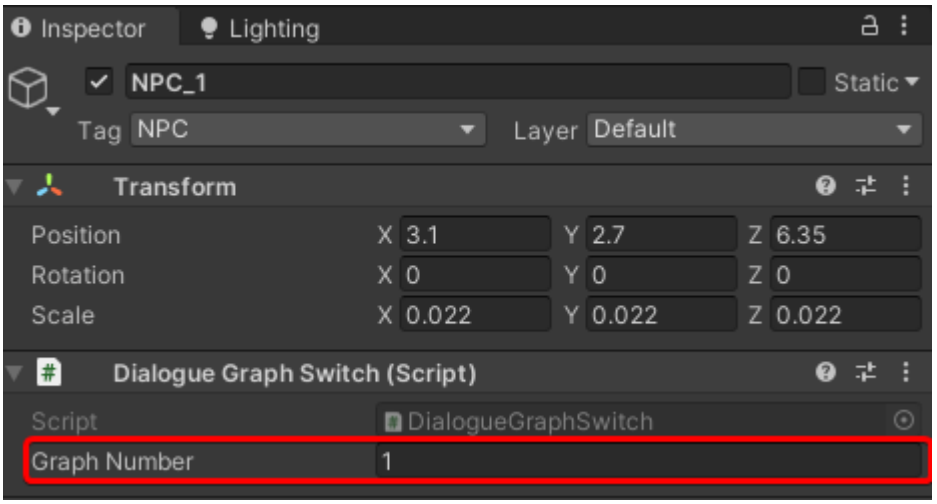
15. An xNode window should pop up and from here, you must first create a Start node and an Exit Node. Then, you create other nodes and connect them. Like this.



16. On the NodeParser script in your PlayerController, drag and drop the DialogueGraphs into the array by setting the amount you want. As shown in red.



17. Go to your NPC and ensure that you set the 'Graph Number' is in sync to what you've added in the Graph array at NodeParser. If you don't do this properly, the NPC won't know which DialogueGraph to use and might identify itself as the first NPC even though the NPC is the second.



Cool tip: If you select a node on the xNode window, you can edit the node text in the Inspector

| Script Glossary

NodeParser - This is where the logic functions of almost every node are defined. It's like the heart of the dialogue system.

DialogueGraphSwitch - As the name suggests, it switches which DialogueGraph to use by accessing the NodeParser's public integer, g, through OnTriggerEnter and finding the Player tag.

InteractionInstigator - It checks if the player interacts with the NPC by pressing 'E' and in close proximity. This script communicates with Interactable.cs

Interactable - Marks the NPC as interactable and communicate with InteractionInstigator.cs

UIInteractionTextController - Controls whether or not 'Press E to Interact' appears OnTrigger. This script is connected to Interactable.cs and InteractionInstigator.cs

BaseNode - Every node is built upon this node. It adds a "virtual" keyword to a function and lets you override in scripts that inherit from this one.

DialogueNode - Adds a basic dialogue node using an override GetString() from BaseNode. Overriding allows you to create a broad type of object that you can refer to but then get specific data from the sub object. It returns a string and communicates back to NodeParser.cs

ChoiceDialogueNode - Functionally the same as DiaogueNode but it communicates with ChoiceDialogeNodeEditor.cs and it uses string lists for choices. It returns a string and communicates back to NodeParser. On the xNode window, writing out the options has a minor bug when you press A or F. Pressing A selects all nodes while F focuses it.

ChoiceDialogeNodeEditor - Mostly controls the appearance of ChoiceDialogeNode when you edit the Dialogue Graph. It adds in more functions in regards to lists.

StartNode - It returns a string and communicates back to NodeParser. The name is self-explanatory.

ExitNode - It returns a string and communicates back to NodeParser. The name is self-explanatory.

DialogueGraph - Allows you to create a Dialogue Graph for your NPC. From there, you create nodes.

CustomNode - It's a placeholder node for you to make your own node. Functionally, it does nothing as you have to code it yourself. You use it by duplicating this script and renaming it while also coding the functions within NodeParser.

| Credits

xNode - Unity Node Editor by Thor Brigsted
https://github.com/Siccity/xNode

Unity XNode Tutorial: Making a Dialogue System by Harrison Gowland
https://youtu.be/7E1EqekXnwQ

Adding a Node Editor - Simple Dialog System Pt.4 (Unity3D Tutorial) by BRAINSHACK
https://youtu.be/ZESlFdSxS58

FirstPersonController
https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2018-4-32351#content

Thanks for the help Max and Nathan :)