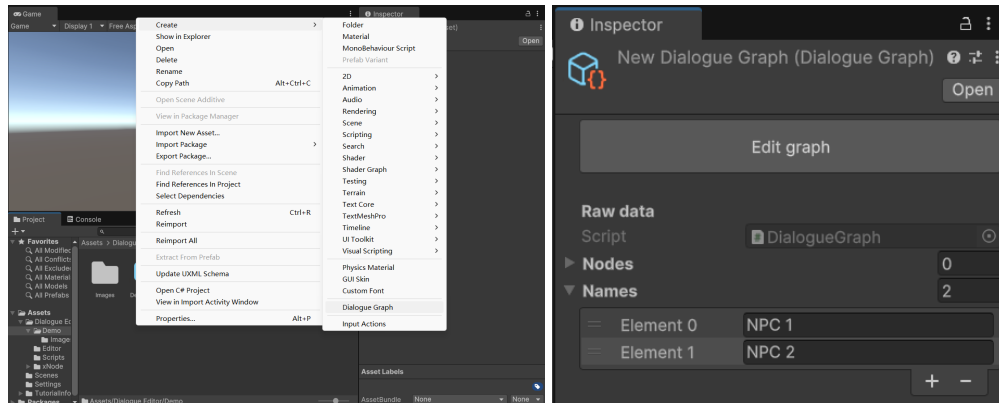


Dialogue Editor

This is a simple visual dialogue editor based on [xNode](#). If you used something like Visual Scripting or Amplify Shader Editor before, you may be familiar with the operations.

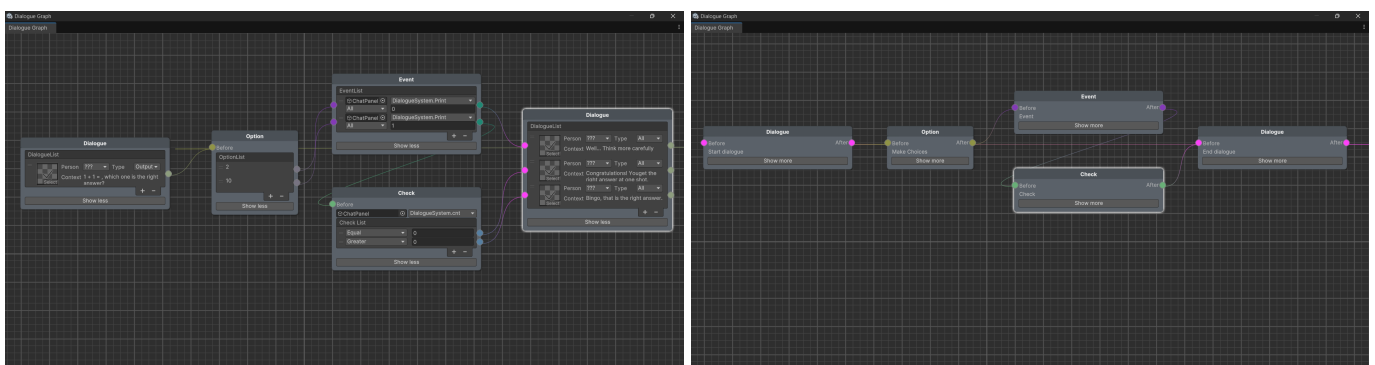
Create

Right click and choose "Create/Dialogue Graph", you will get a file named New Dialogue Graph. Before we double click the file to edit it, we can add name to the file's list in the inspector so that we can dropdown to select name in dialogue node later.



Edit

There are four different kind of nodes in Dialogue Editor. They are dialogue node (the most commonly used one), option node (making choices is also usual in games), event node (Invoke some public functions), check node (check a public variable, different value will lead different direction). The items in dialogue node and event node can have both input port and output port, this makes the node more flexible. Overall the flow of information is clear, only one thing needs to be taken care of is that if one item have an output port connect nothing, then it is an end point. By the way if the node is too long, you can also press the "Show less" button, it will save the connection info and show all of them.



Use

You need to load the graph you make as an scriptable object, then instantiate to use it. Generally, you will only use the chatInfo, optionInfo, and Next(). Though the Next(), you can get the type and relative information of the current step. And the you can get the corresponding information though dialogueInfo or optionInfo.

```
//Variables and functions
public enum DataType { End, Dialogue, Option }
```

```

public struct DialogueInfo { public Sprite sprite; public string name, context; }
public DialogueInfo dialogueInfo;
public List<string> optionInfo;
public DataType Next(int num = -1) // -1: continue dialogue,
                                   // 0 ~ inf: option's index that choiced

//Useage
public ScriptableObject graph; // Load your graph file
DialogueGraph dialogueGraph = (DialogueGraph)Instantiate(graph);
switch (chatGraph.Next(num))
{
    case DialogueGraph.DataType.Dialogue:
        YourSprite = chatGraph.dialogueInfo.sprite;
        YourText = chatGraph.dialogueInfo.name + ":\n" + chatGraph.dialogueInfo.context;
        break;
    case DialogueGraph.DataType.Option:
        for (int i = 0; i < chatGraph.optionInfo.Count; i++)
            YourOptions = chatGraph.optionInfo[i];
        break;
    case DialogueGraph.DataType.End:
        Do Something...
        break;
}

```