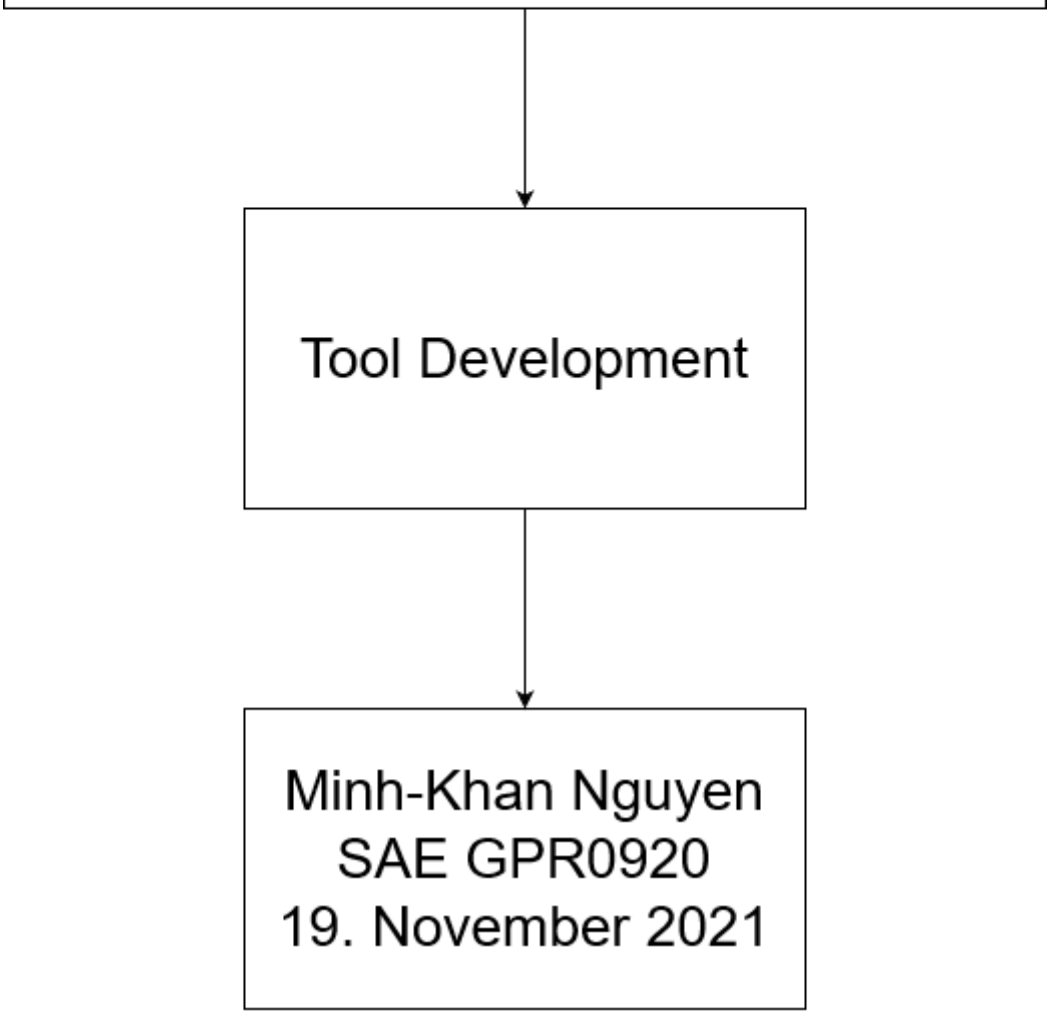
EASY Dialogue System TOOL PRO+ 2021 (new)



Minh-Khan Nguyen
SAE GPR0920
19. November 2021

Abstract

It’s a visual, node-based dialogue editor.
[Under construction]

Introduction

There are already similar tools with many features and years of development but as a challenge for myself, I've set the goal to implement an own Dialogue System Tool with the bare minimum and make it available for free.
The title or name itself is a jab at all the amusing and creative tool names in the asset store.

Related Work

The popular market leader is the Dialogue System for Unity by Pixel Crushers.
It has more functionality and options than Unity itself, also node-based and extremely versatile. The price is reasonable, but leans slightly towards a higher price range.

Features

Must have

- Nodes / Visual editor for a simple and flexible handling of the conversations
- A minimalistic and clear design of the textbox, font
- An ID for each box for backend interactions and oversight
- A variety of interactable buttons
- Drag input for speaker and listener actor
- Transition, conditions and events

Nice to have

- Free control of the design for the textbox, font and buttons
- Text to speech
- Output of audio, noise or visual cue

Mock Up

Dialogue Window Design

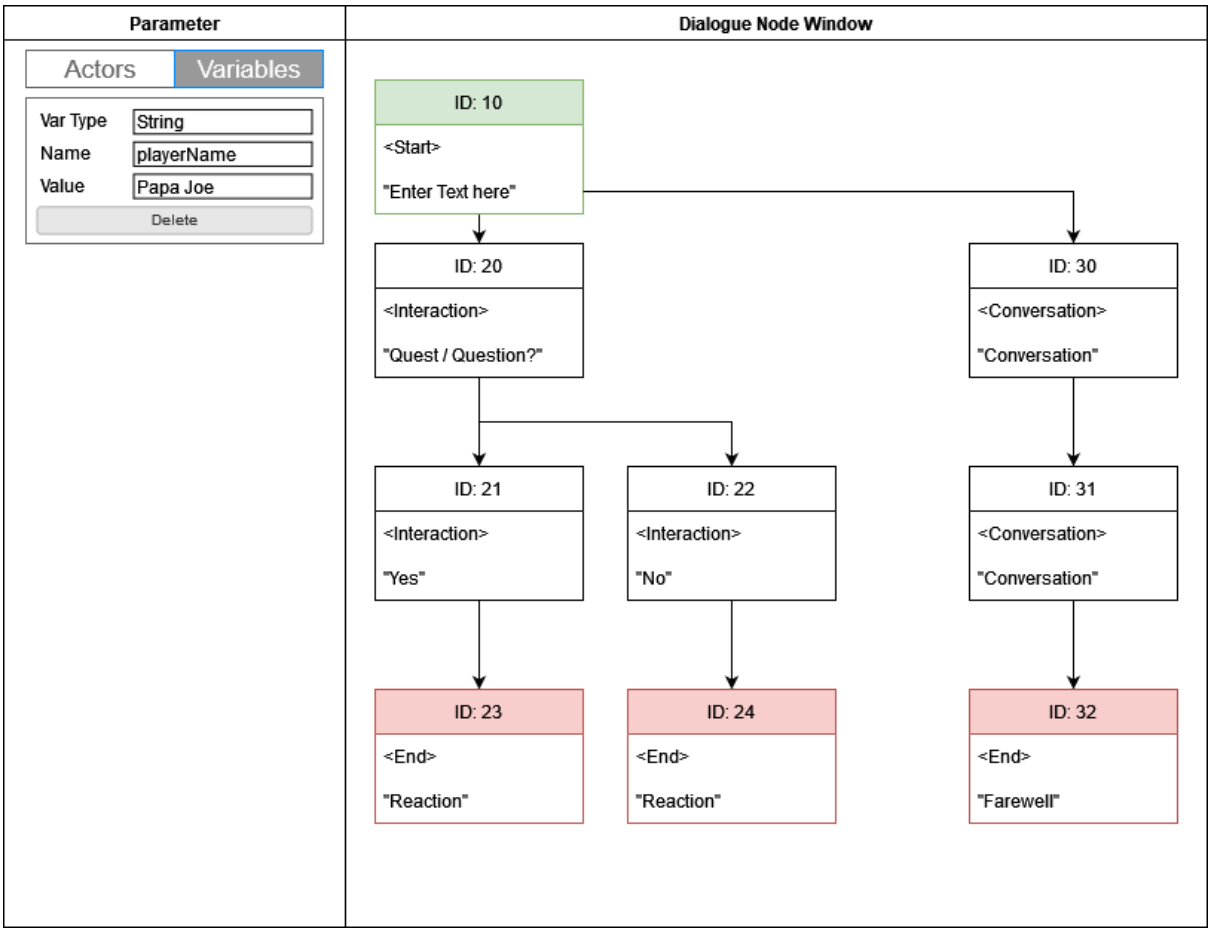


Figure 1: User Interface of the Node Window

The window can be called inside the Unity Editor on the tabs.
You can create new speech bubbles with right click anywhere in the Node Window and assign their ID, the interactions and the text in the inspector.
Right click on the nodes to connect them with the straightforward node system and save it as a scriptable object to give the actor their own conversation branch with the flexibility to save and expand it anytime.

In figure 1 is a parameter window visible, inside of that window, you can create new variables and actors and assign them later on in the inspector in Figure 2.

Inspector Window Design

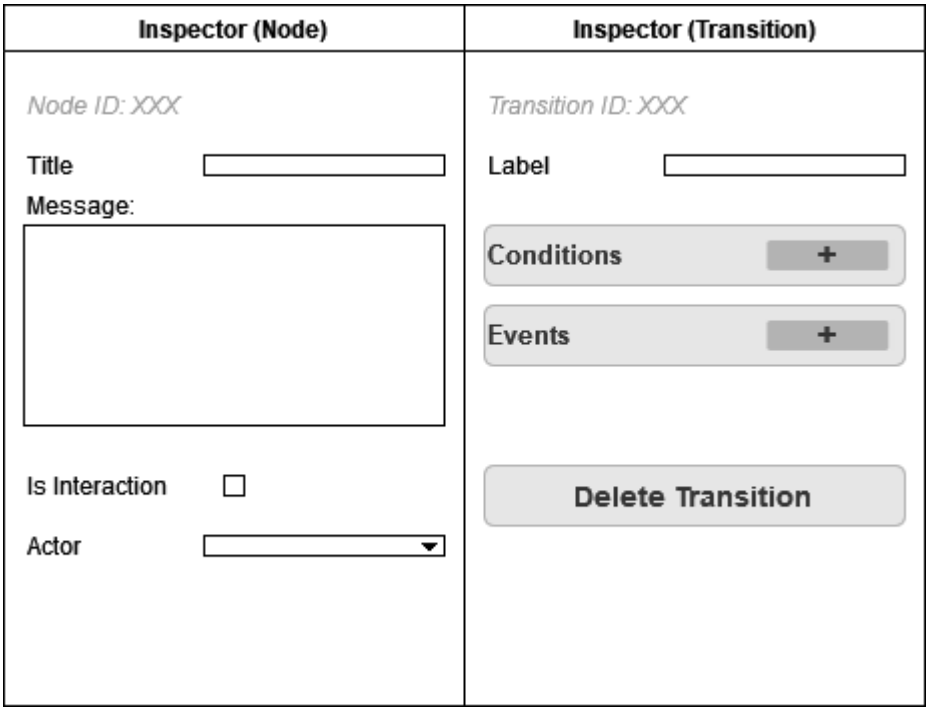
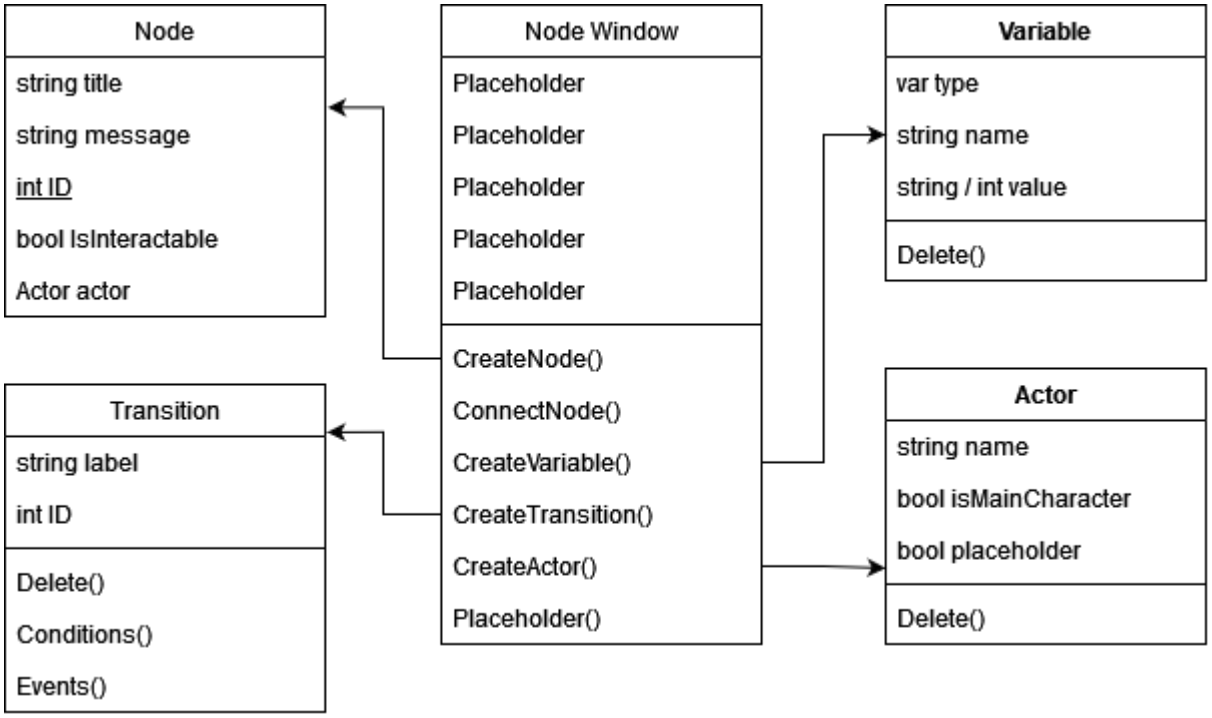


Figure 2: User Interface of the Inspector (Nodes)

In the inspector on the left you can find and edit the title, node ID, message, interactable and the actor if you click on a node.
On the right you can edit the label, transition ID, conditions and events for the transitions (arrows connecting the nodes).
Condition values are created in the Dialogue Window and set in the transition inspector, executing events like proceeding with another script or node.

Class Diagram



Game Example

Figure 3 : Conversation in The Legend of Zelda: Breath of the Wild
[Will be replaced or compared to own in-game example]

In figure 3 is a proper dialogue design visible, functioning as a template for my own implementation and development of my dialogue tool.

Timeline

	Task Name	Duration	Start	ETA
1	Finish Tool	28 Days	20.11.21	16.12.21
2	Finish Tool Prototype	20 Days	20.11.21	9.12.21
3	Node programming	10 Days	20.11.21	30.11.21
4	Adding values / actors	2 Days	1.12.21	2.12.21
5	Configure inspector and variables	2 Days	3.12.21	4.12.21
6	Conditions and events trigger	3 days	5.12.21	8.12.21
7	Implement reusable data containers	2 Days	9.12.21	11.12.21
8	Self-testing and bug fixing	4 Days	12.12.21	16.12.21
9	User Interface / Design	7 Days	9.12.21	16.12.21
10	Simple UI	2 days	9.12.21	10.12.21
11	Update GDD of tool	1 Day	15.12.21	16.12.21
12	Set up Github repository	< 1 Day	20.12.21	20.12.21
13	Cry	7 Days	9.12.21	16.12.21
14	Consumer and User feedback	2 Days	14.12.21	16.12.21

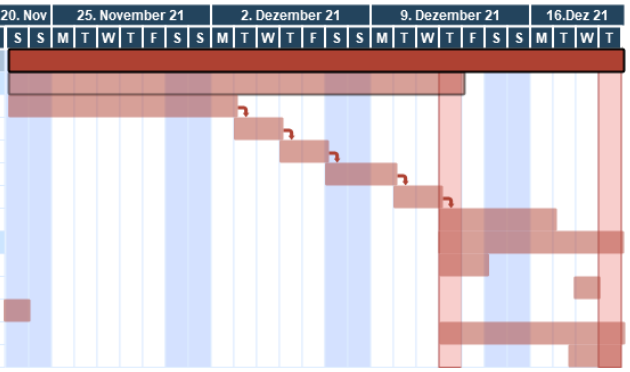


Figure 4: Timetable

This is a short summary of the project, for the full timeline check it out on the Hack'n'Plan website.

Compatibility

System

- Windows 10 (tested)
- MacOS (not tested)
- Linux (not tested)

Platforms

- Unity 2020.3.18f1 LTS

Links

[Under Construction]

Conclusion

[Under Construction]