**Event Generation Tool**

**Table des matières**

[**1. Introduction 2**](#_Toc502677578)

[**2. The Node Editor Window** 3](#_Toc502677579)

[**a.** **Open the editor** 3](#_Toc502677581)

[**b.** **Create a new node** 3](#_Toc502677585)

 [**c. Event nodes** 4](#_Toc502677588)

[**d. Event Actions** 4](#_Toc502677589)

[**e. Events Outputs** 5](#_Toc502677590)

[**f. Types of inputs** 5](#_Toc502677591)

[**g. Make a transition** 5](#_Toc502677592)

[**h. Save the flowchart** 6](#_Toc502677593)

[**3. How to create a flow in the game scene** 7](#_Toc502677594)

[**a. Scene Setup** 7](#_Toc502677596)

[**b. Load the flow** 7](#_Toc502677597)

[**c.** **Event Properties** 8](#_Toc502677598)

[**d.** **Actions in Events** 9](#_Toc502677599)

# **1. Introduction**

This tool aims at creating a simple way to prototype and develop game events in a scene for the designers without having to add a single line of code.

# **2. The Node Editor Window**

* 1. **Open the editor**

We will proceed step by step, first, in unity go to the **Window tab** and click on **Node Editor**



* 1. **Create a new node**

A new window opens, here you can **right click anywhere** and an option “Add Node” will appear. Click on it to create a new node.



*Nb: You can* ***left click on a window to and drag where you want it to be****. You can also* ***maintain the middle click and drag to navigate*** *in the editor window, there is no way to Zoom-in/out now so it is the only way to show more nodes than your screen can handle.*

### **c. Event nodes**

A node is made of:

• TITLE:

This is **how your event will be called**.

• NUMBER OF ACTIONS:

All the **actions that will be triggered** when the event is loaded ingame.

• NUMBER OF OUTPUTS:

All the other **events it is linked to**.

## **d. Event Actions**

Once you’ve created a node and renamed it, **you can add actions** to it. **These actions will be executed once when the event is triggered during gameplay**. To do so, **click on the little circle next to the “Action On Trigger” fields**.





It will open a new window in which you can **select the actions you want to add** to the event. You can only put **one per field**. If you want **to add more, change the “Number of Actions” field** **to something bigger.**



## **e. Events Outputs**

Once you’ve added your actions, your node should look like that. Now it is **time to add some outputs**. It works **almost the same way as the actions**.

The **“Number of Outputs” field determines the number of events you want to go to**. It is composed of two fields “**Input**” and “**Output**”, the **output field will be generated automatically** so **no need to worry** about it.

 If you click on the input, a dropdown list will appear, **each object in it corresponds to an action ingame**.

## **f. Types of inputs**

• NOD: Player has to do **Yes or No**.
• POINT: Player has to **point an object** in the scene.
• LOOK: Player has to **look at an object** in the scene.
• YES: Player has to do **Yes**.
• NO: Player has to do **No**.
• EXTERNAL: The event will be **triggered by a game object in the scene**.
• IDLE: The event will be **triggered after a TimeOut**.

## **g. Make a transition**

Once you’ve created several nodes. You can **right click on the white square** next to the output field, a little box **“Make Transition” will appear**. **Left click** on it **to start the transition process**.



A red curve will appear, you can **left click on the square on the left of an event to link the output**.

And that’s it! Now **you can create your whole flow of events**. **All events do not have to be linked** to be created ingame.

**If you want to create different loops within the same state machine, you can**.

## **h. Save the flowchart**

Once you’re done creating your flow, in the “**Events Folder**” field, **write the name of your scene**.



Then **hit the “Save Current Flow” button to save your flow** as an asset you can load.

If you want **to load a previously created flow**, put your **save asset in the “SaveFile” field** and hit the **“Load Flow” button**.

**To go to the next step of the creation process, hit the “UNWRAP FLOW” button, it will create a folder filled with event files that will be needed afterward**.



*Nb: The event files will be saved in Assets/Resources/FolderName*

**3. How to create a flow in the game scene**

**a. Scene Setup**

**Create a new game object** and attach the **NPC\_EVENT\_HANDLER** component to it.

**b. Load the flow**

**In the “Event On Trigger”** field, **add the first event of the flow you created**. Then **hit the “Do Everything” button**.

****


It will generate a new GameObject for each event of your flow.

* 1. **Event Properties**



Each event object created will receive all the actions you’ve attributed to them in the node editor and is also given a EVENT\_PROPERTIES component.

First let’s take a look at the event properties.

CONNECTIONS: For each output you’ve created in the node editor, a field will be added in the list with the required input, the event it is linked to. There is a new field “Object to Output”, it is in fact the object you need to interact with to validate the input (Ex: The player has to look at a tree to go to the next event, then drag the tree in the object field).

HAS CONDITIONS: Tick this box if you want to reroute the player to a different event if he has already completed a particular event. If you do so, a new list will appear, click on the “+” to add a new condition.

CONDITIONS: There are two fields per condition, “If this event triggered” and “Go to this event”. The first one will check if the event you want to check has already been through, if yes it will go directly to the event you’ve linked in the second field.

 (Ex: Node 3 & 4 are linked together but you want to go to node 5 if you’ve visited both. Then in Node 3 tick the has condition box, drag the Node 4 Game Object from the scene to the “If this event triggered” field and drag the Node 5 asset from your project (Resources/FolderName/) to the “Go to this Event”)



* 1. **Actions in Events**

Now let’s give a look to the actions. Each of them has a particular action but they work the same way. Once an event is triggered, all the actions in it will be read one by one in the order they are in the inspector (from Top to Bottom).



HAS TIMER: Tick this box if you want the actions to be executed after a delay.

NEED TIMER BEFORE NEXT ACTION: Tick this box if you want to delay the execution of the Next action.

If you tick either of these two boxes, a new timer field will appear for each, it’s where you will have to put the time in seconds you want to wait.

