

First, thank you very much for purchasing this asset, and I hope it saves you a lot of time developing your projects.

If you find any errors or have suggestions for additions to the asset, please complete this form:

<https://docs.google.com/forms/d/e/1FAIpQLSeknvoOKogqmuhqEdAAS8ACpKuCa1or-fMnp0Ym5GPvTKh2Ng/viewform?usp=sharing>

Original Unity version: Unity 6 (6000.025f1)

It is perfectly compatible with previous versions

Required packages:

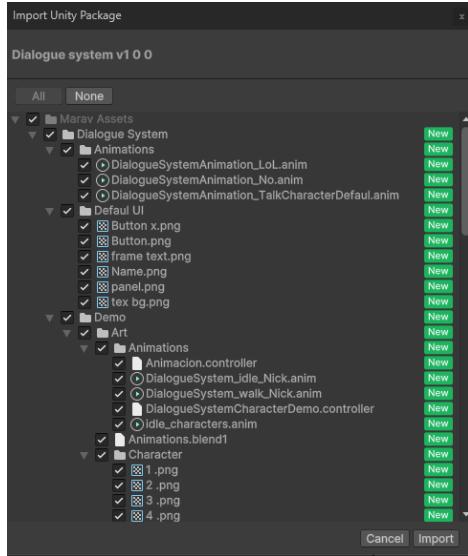
- **TextMesh Pro**
- 2D Sprite (Demo only)
- 2D Animation (Demo only)

— Start here —

How to make a “Hello world”:

“Hello world” is a phrase used in programming to show on the screen that a program is running correctly. In our case, we are going to have a character say hello world in 2 languages.

Step 1: We import the asset



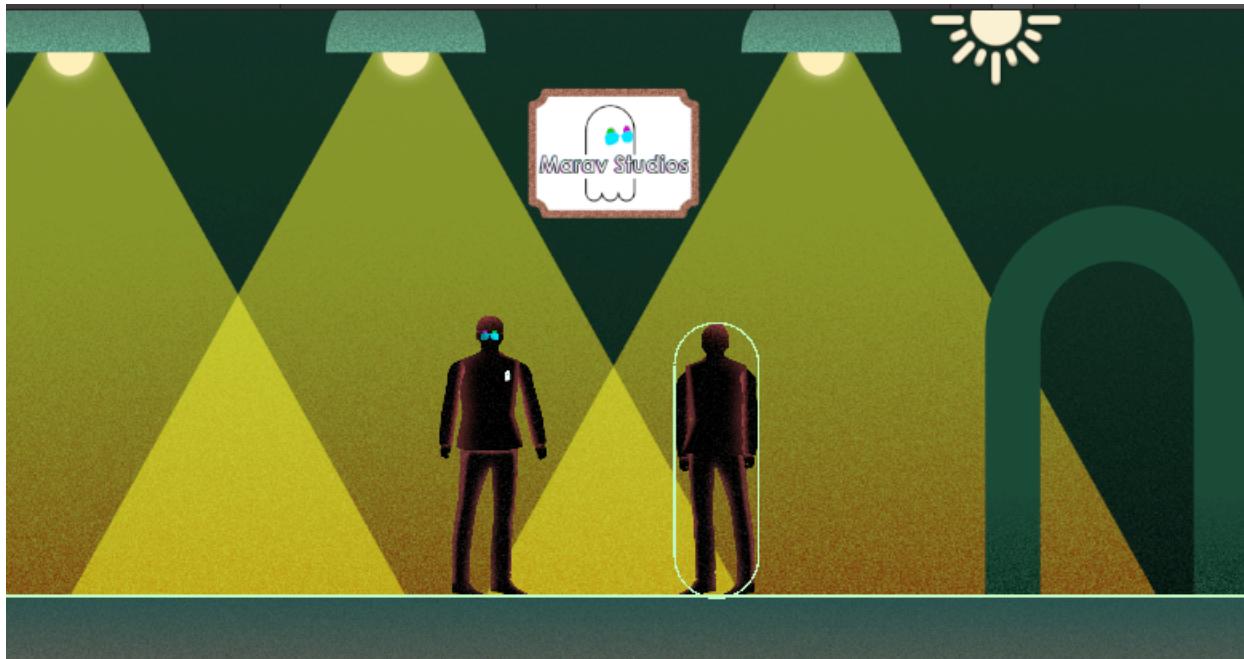
Make sure the following packages are installed beforehand:

- TextMesh Pro
- 2D Sprite (Demo only)
- 2D Animation (Demo only)

Step 2: We set the scene

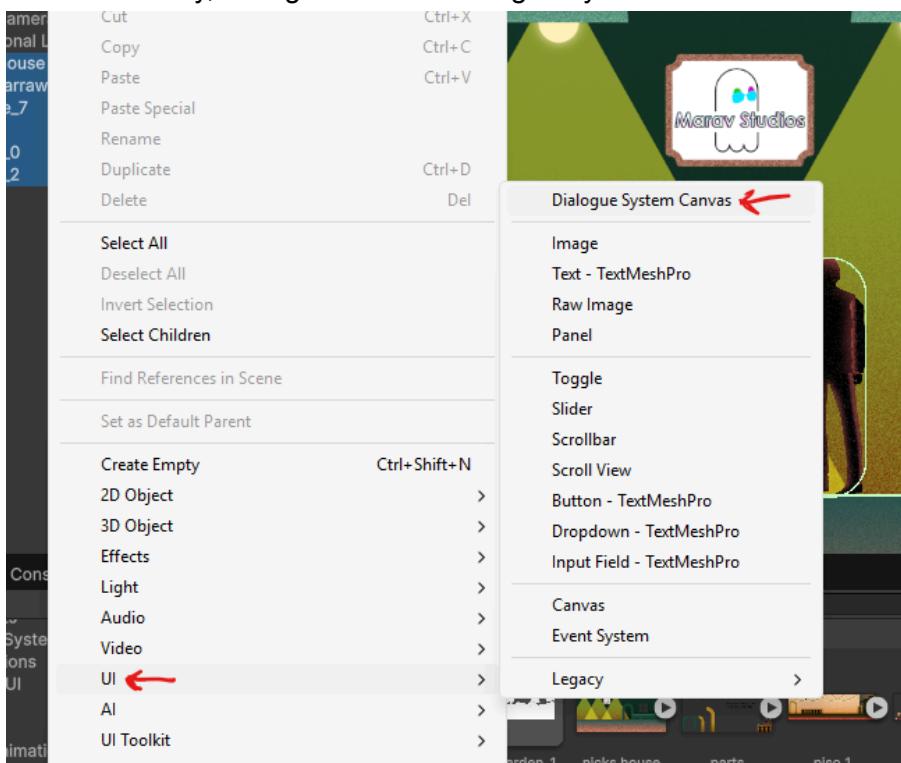
The scene must have:

- The player, with his “Player” tag
- An NPC, the one who will speak

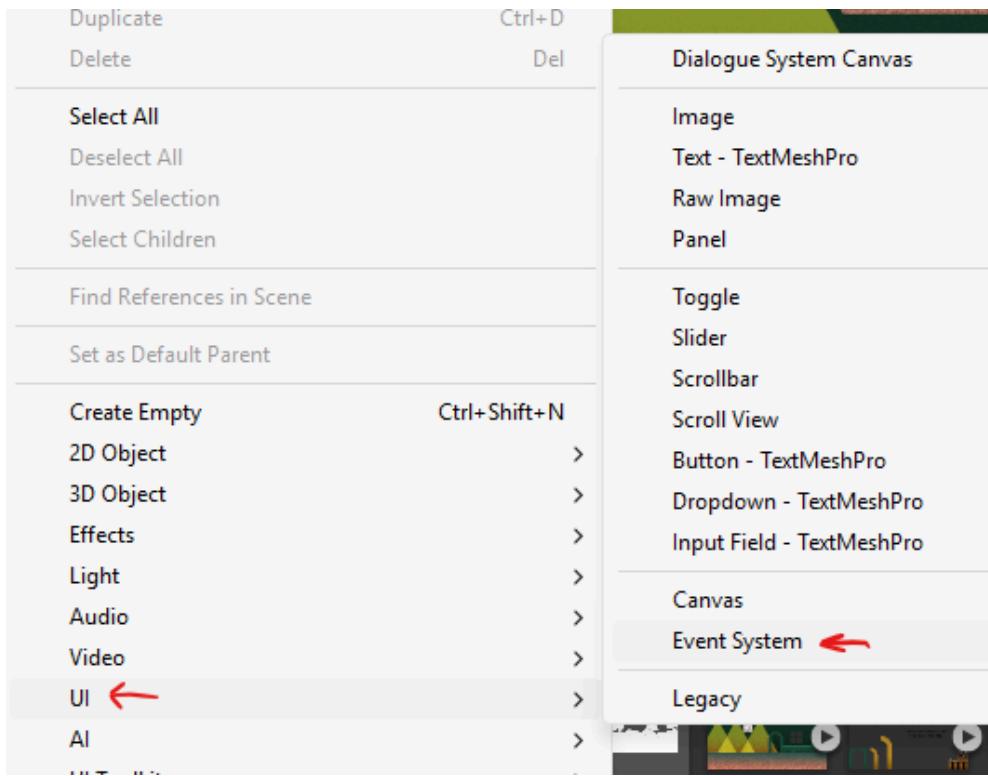


Step 3: we prepare the Dialogue system interface:

In the Hierarchy, we right click UI/ Dialogue system Canvas



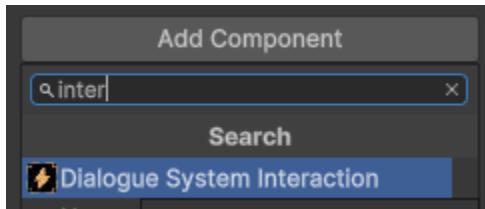
If it is not in the scene, we create the “Event System”. For this we right click UI/ Event System



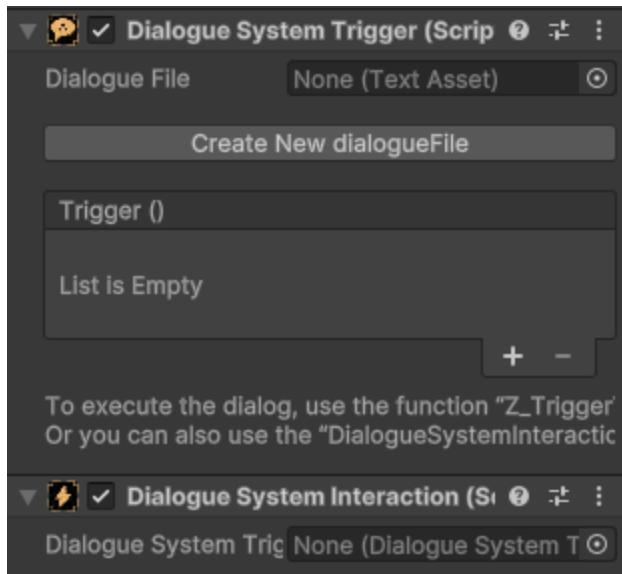
This completes the UI setup.

Step 4: We configure the NPC

For this, we go to who will be the NPC and add a Dialogue System Interaction component



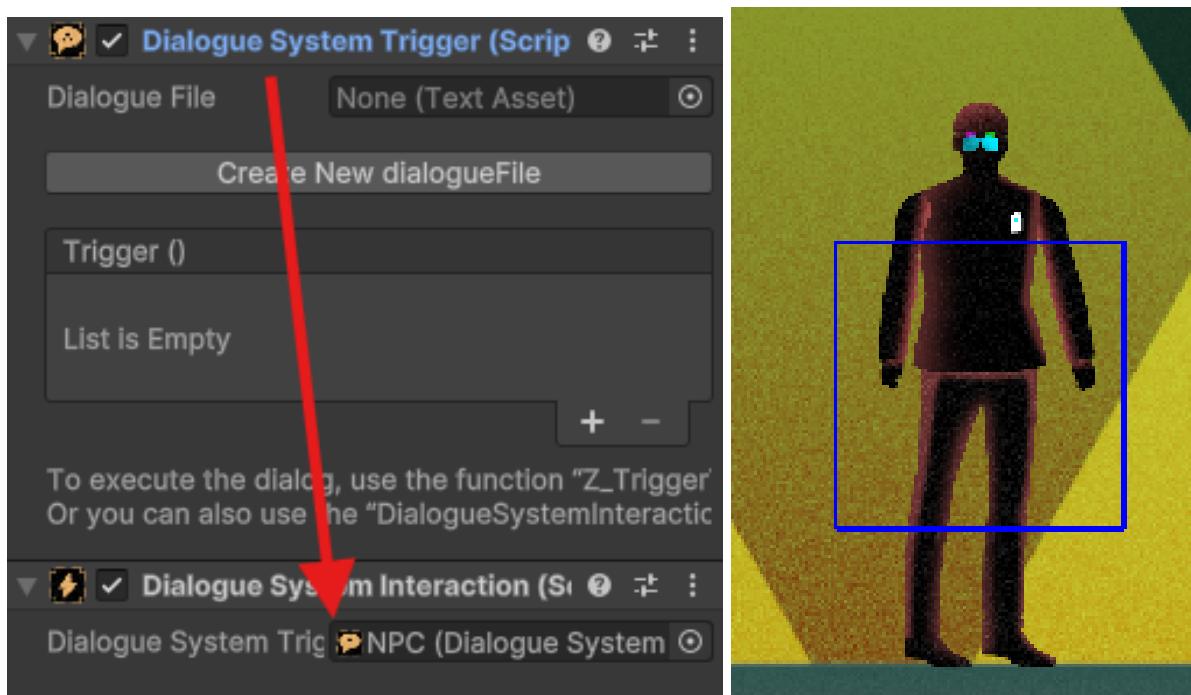
This will add two components, the Dialogue System Interaction and the Dialogue System Trigger:



The Dialogue System Trigger executes the dialog.

The Dialogue System Interaction executes the Dialogue System Trigger. I recommend making your own code that runs the Dialogue System Trigger, as each project will require different ways to run a dialog.

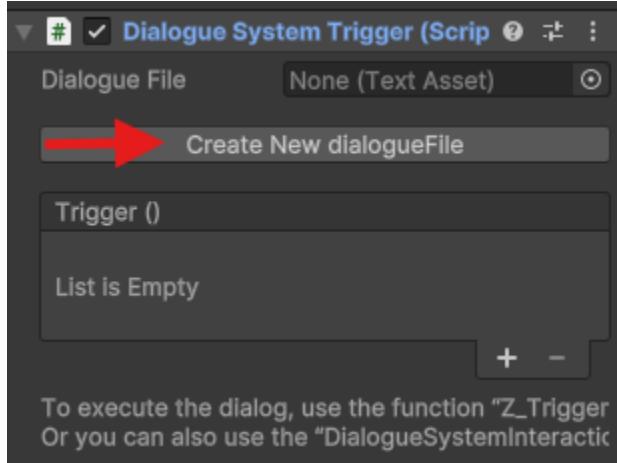
We assign the Dialogue System Trigger to the Dialogue System Interaction. Next we will see how the latter is activated and projects a collision frame on the scene.



If we are working in 2D, we already have it ready.

Step 5: Create a DialogueFile

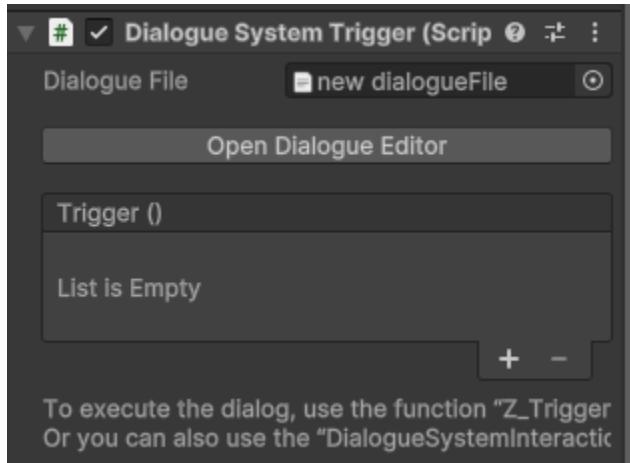
We click on the “Create New DialogueFile” button



It will ask us to give it a name and location, we have to save the file within the project.

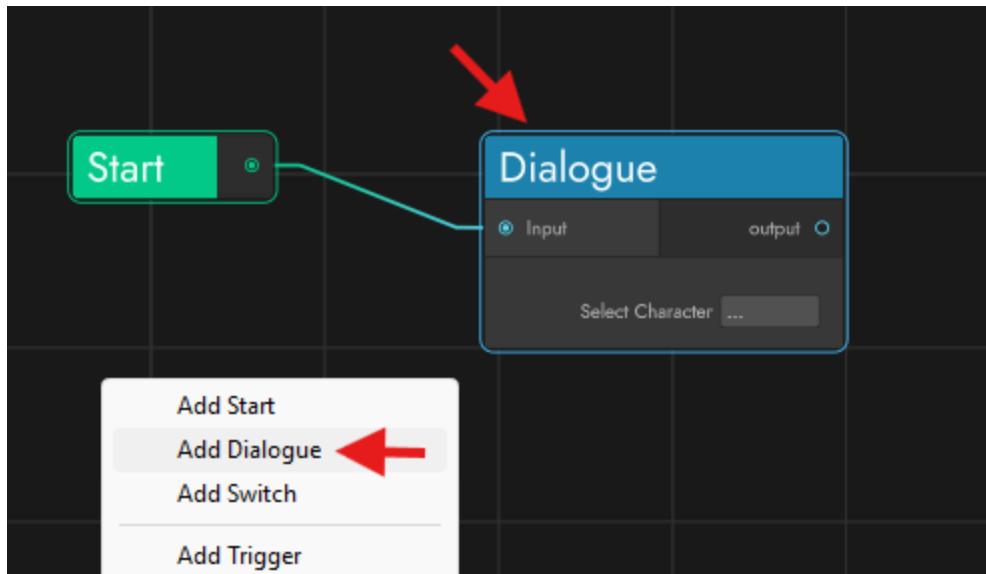
Step 6: Edit the DialogueFile

Once the DialogueFile is created, click on the Open Dialogue Editor button

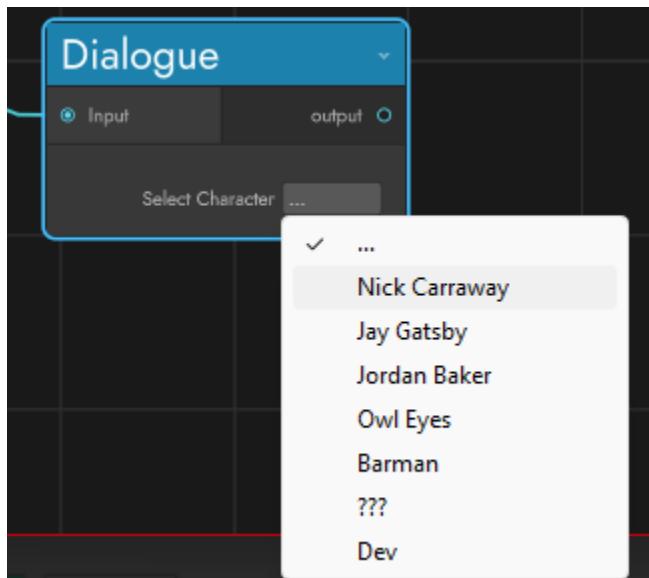


The node editor will open:

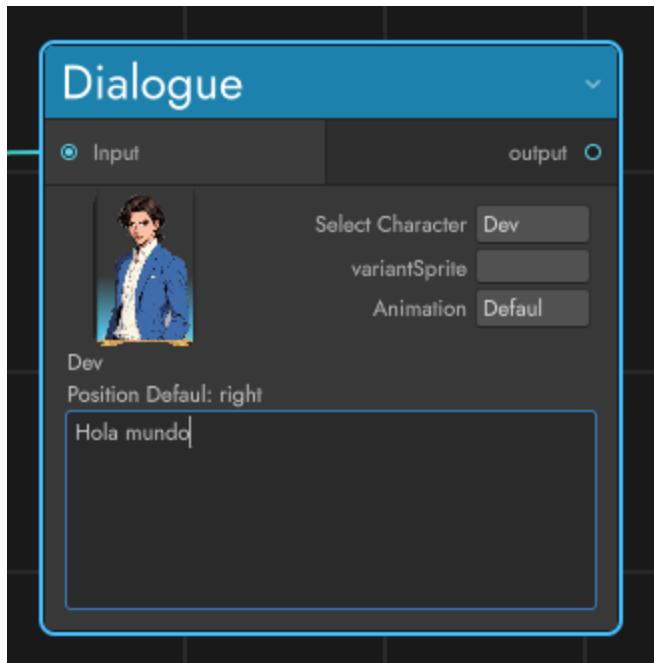
Now, right-click to create a new node and select 'Dialogue Node'. Then we connect it to the start.



We select a character, in my case, "Dev".

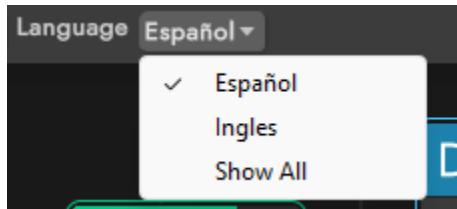


We put hello world in the text box

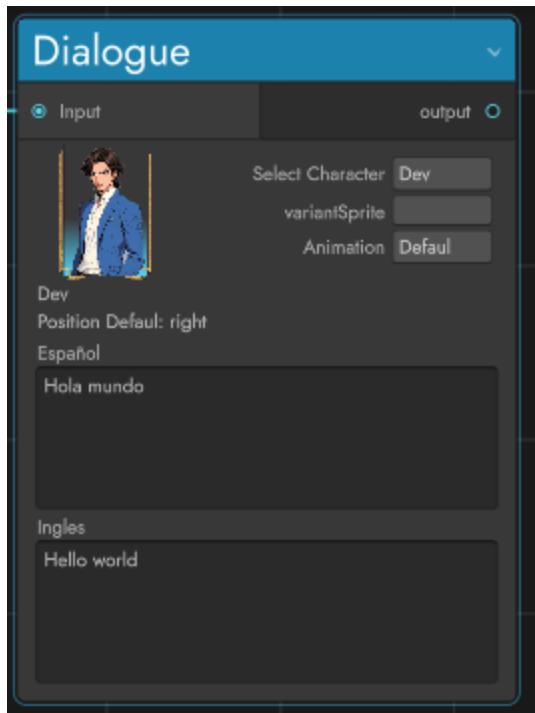


Step 7: Second language

All languages will appear in the top bar. I recommend doing the translations with the Show All option

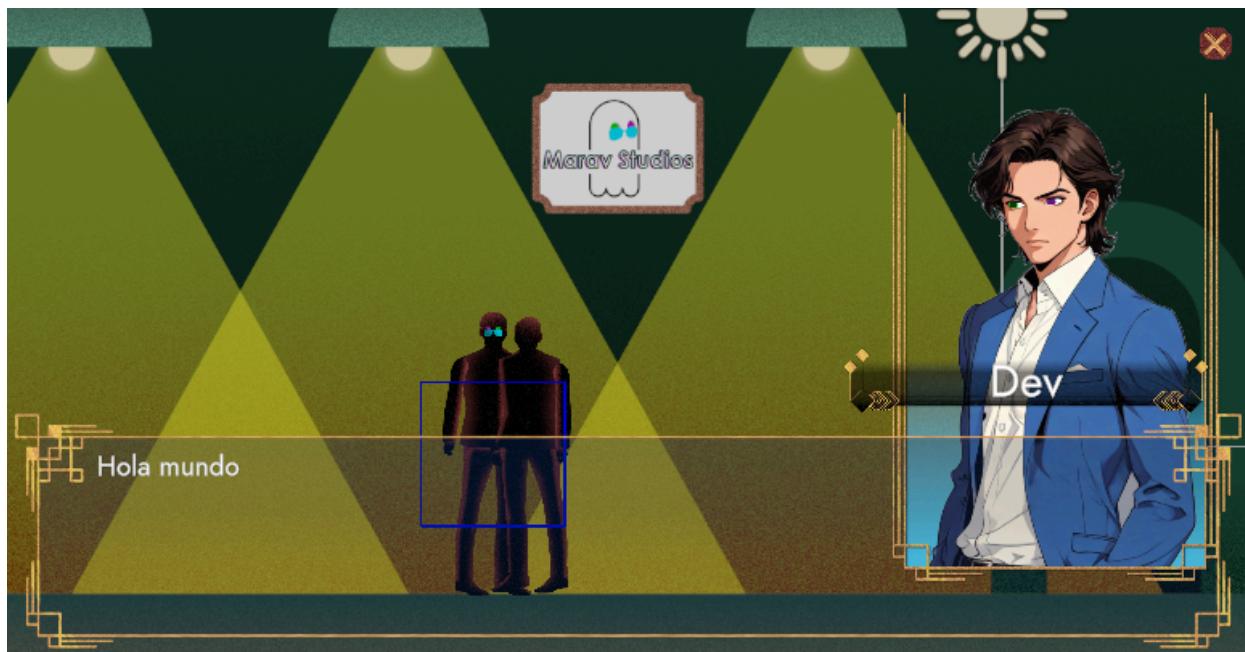


With 'Show All' enabled, multiple text boxes will appear, one for each language.



We click on the Save button to save correctly

Step 8: We test



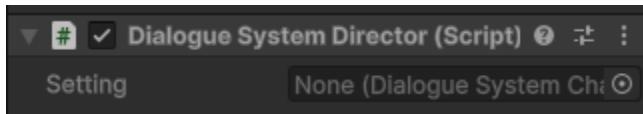
Success!

Common problems

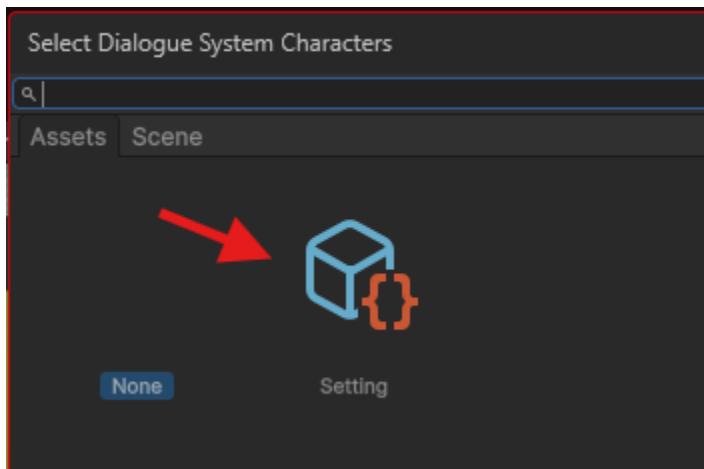
1. This error:

```
[15:50:00] NullReferenceException: Object reference not set to an instance of an object  
DialogueSystemDirector.Start () (at Assets/Marav Assets/Dialogue System/Script/DialogueSystemDirector.cs:78)
```

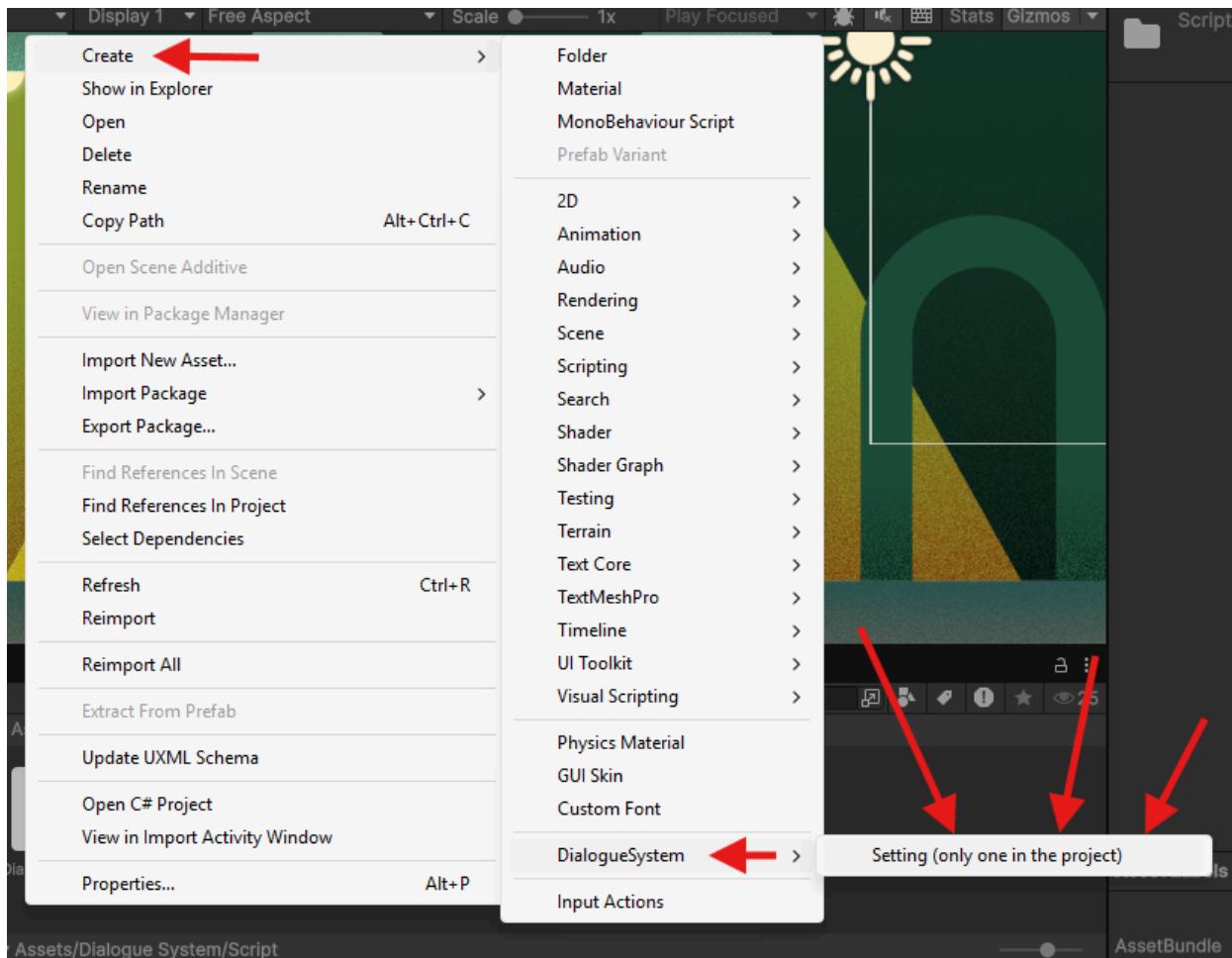
If it runs as soon as the project is put into Play, the problem is with the director
You simply have to select the Setting



It is recommended to have one in the project.

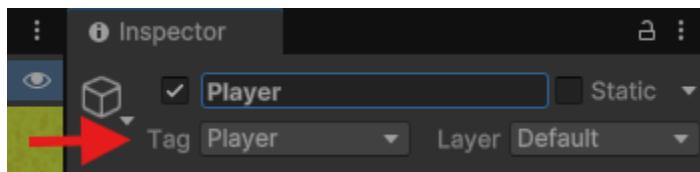


If none exists, create one.

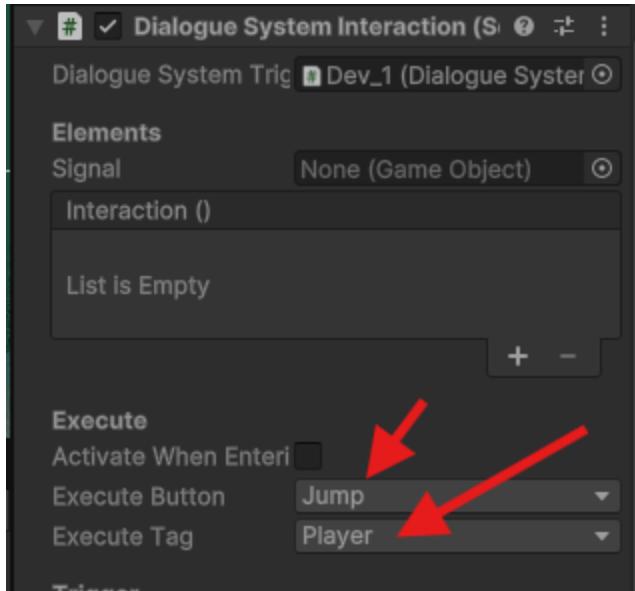


2. The player does not execute it:

Remember to put the Player tag, to the Player



Also check the Dialogue System interaction, by default it is executed with the "jump" button, which corresponds to the space.



If you have problems, feel free to fill out this form:

— How it works —



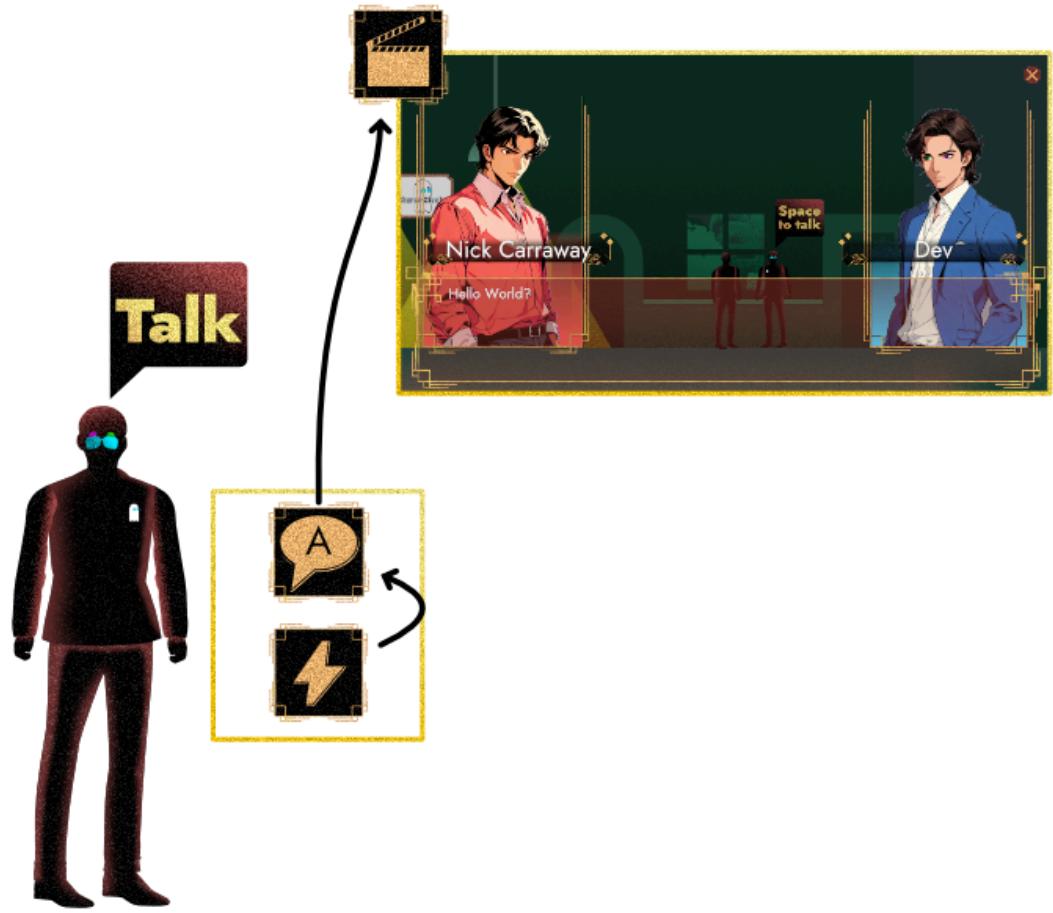
Dialogue System Interaction: This is responsible for interacting with the world of your project. Its role is to detect when to start the dialogue. It is separate from the Dialogue trigger because each project will need to run the dialog on different occasions, with different parameters.



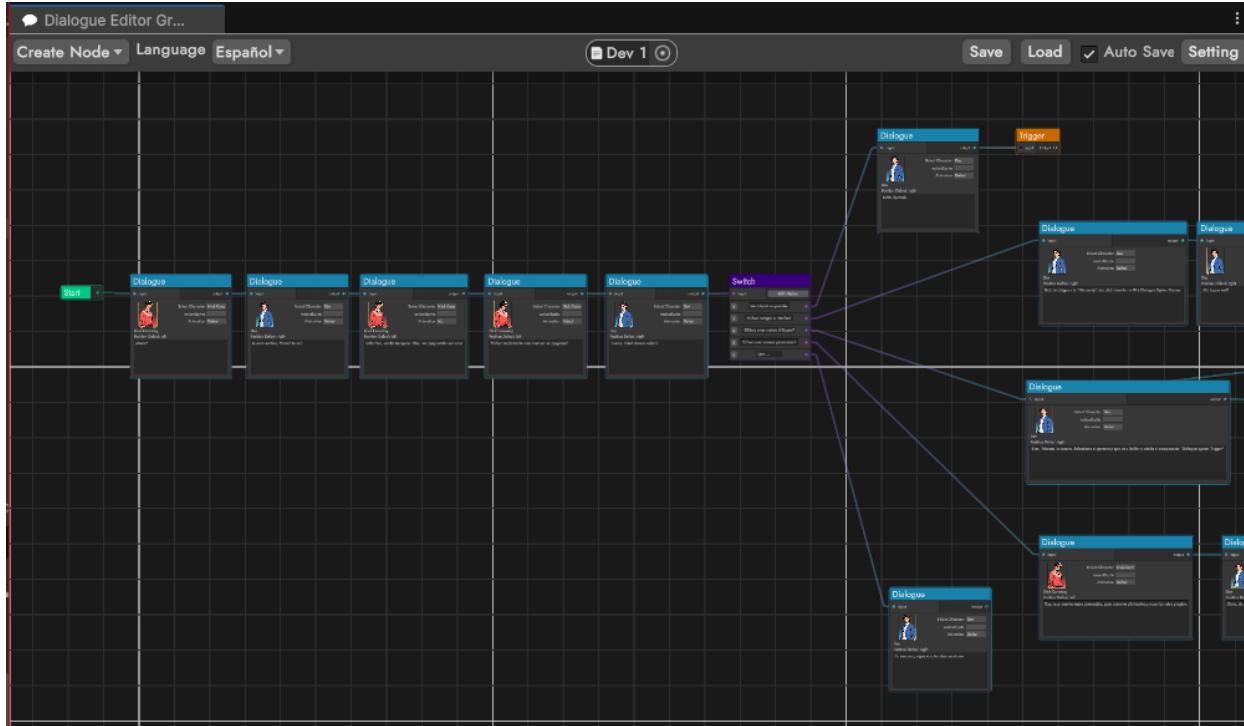
Dialogue System Trigger: This is responsible for executing the dialogue. It is as if an assistant passed the script (Dialogue file) of the scene to the director of a movie.



Dialogue System Director: Responsible for reproducing the scene. When a Dialogue System Trigger passes it the Dialogue file, it will play it.



Completely independently there is the “Dialogue Editor Graph”. This is responsible for editing and writing the Dialogue Files.

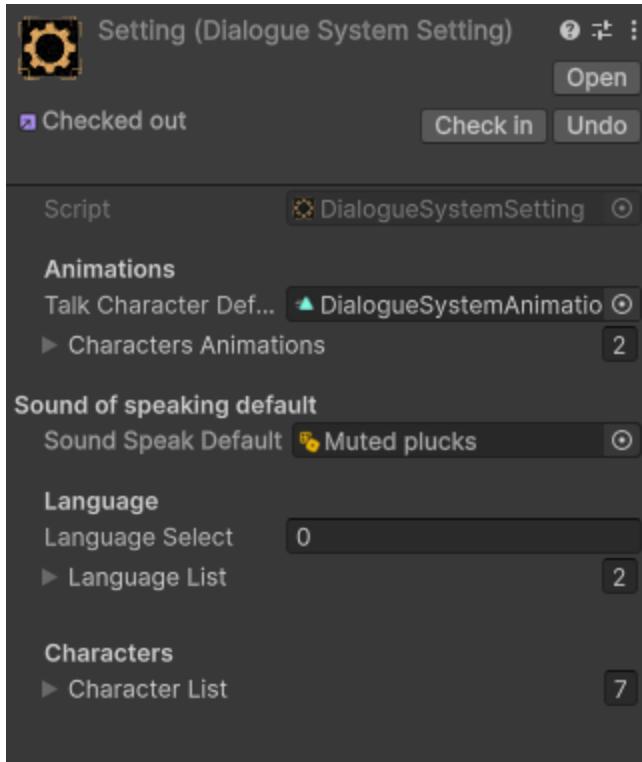


Finally, the general information of the project must be saved somewhere, such as the characters, the available languages or the animations. This is what the Dialogue System Setting is for to save all the general information of the dialogues.

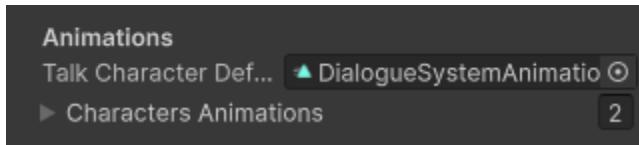
— Components and objects —

Panel de setting

All the general information about the behavior of the Asset is saved here, I recommend editing everything. From start to finish.

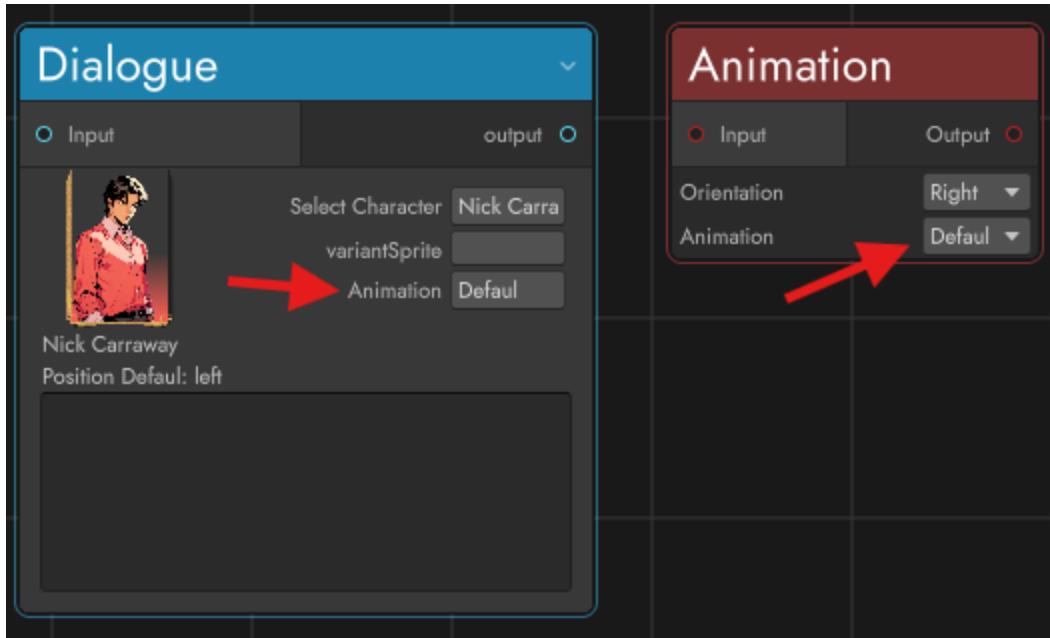


Animations



When a character speaks, he or she executes an animation. You can have different animations run depending on the case.

- Talk Character default: the default animation of the characters when speaking is saved there
- Characters animations: the rest of the animations are saved there. Then these appear in the node animation selector

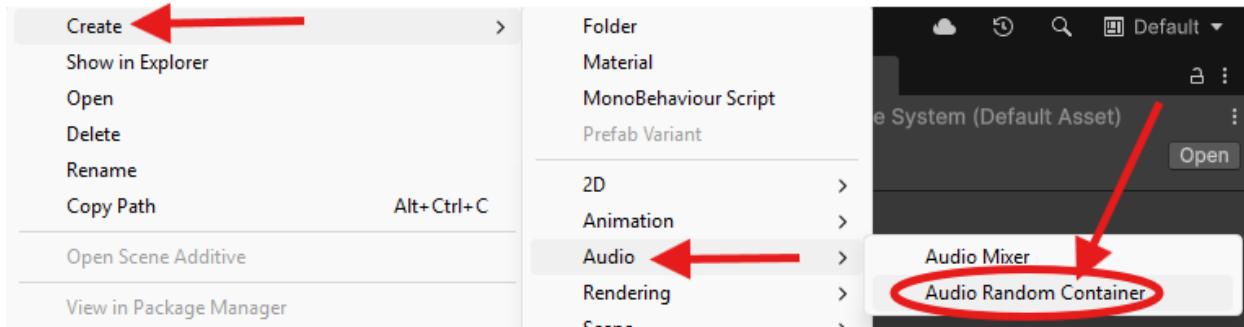


Sound of speaking Default

To give better feedback, when a character speaks, a sound is played to make it sound like the character is speaking.



It is important to use an Audio Random Container.



This tea will give the possibility to add variations, to break the monotony and give a better immersion.

Language

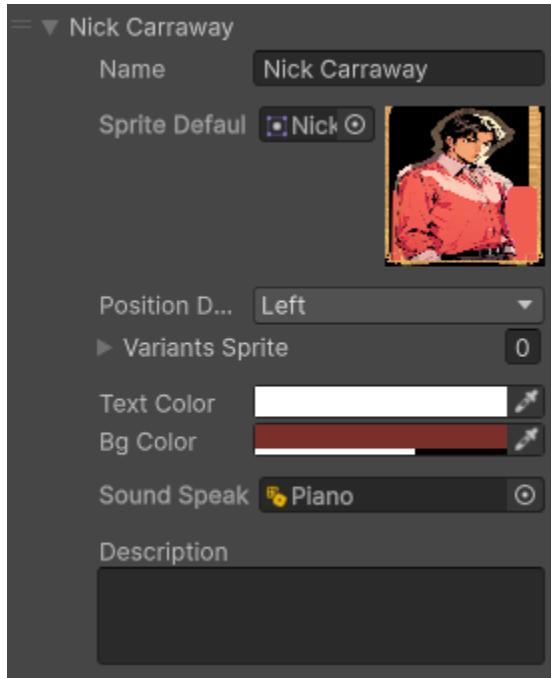
This records the languages that will be used in the project. Its effect is directly reflected in the language selector in the editor.



- **language select:** The language that is currently in use
- **language List:** the list of available languages.

Character

The information about the project characters is saved here.

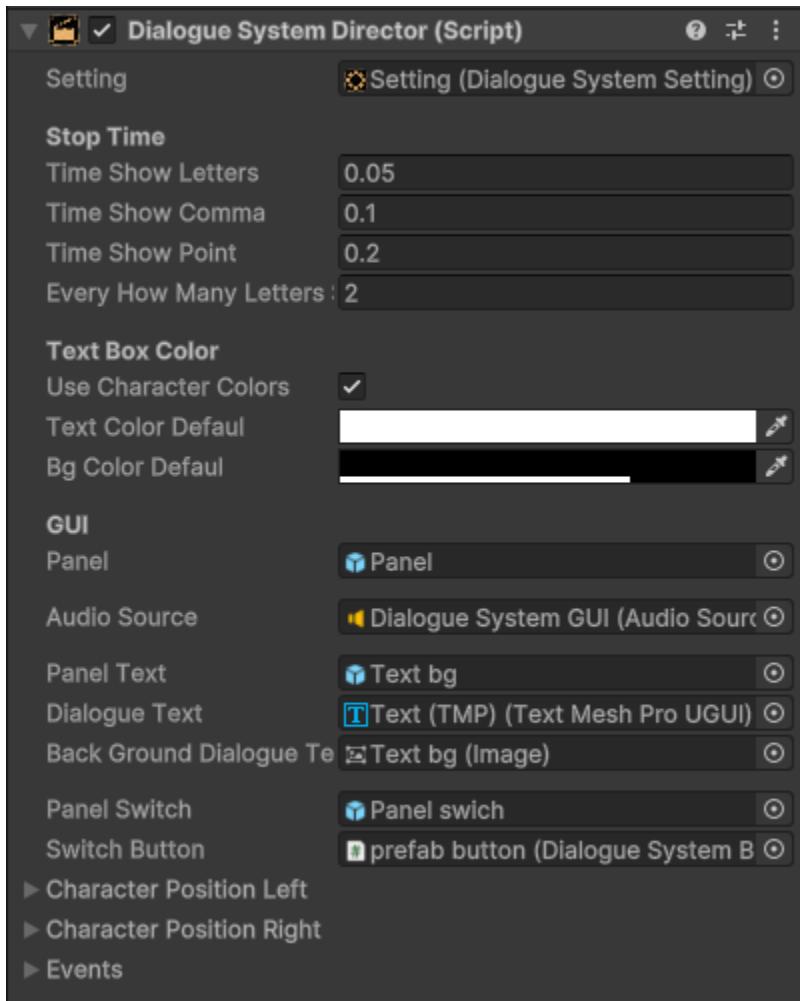


- **Name:** Name of the character.
- **default sprite:** This is the default sprite that will be displayed when speaking.
- **position default:** Refers to the default position on the screen, whether it appears on the left or right.
- **variant sprite:** Different sprites can be displayed depending on the case. Normally, it is used to represent emotions.
- **Sound Speak:** The sound it plays when speaking, if left empty, uses the Sound Speak Default.

- **text color:** Refers to the color of the text when speaking. (this function can be disabled if desired).
- **bg color:** Refers to the color of the background of the text when speaking. (this function can be disabled if desired).
- **Description:** A brief description of the character, it has no function beyond that.

Director

This executes the Dialogue file, as if it were the director of a movie, organizing and executing the script (dialogue file).



Stop Time

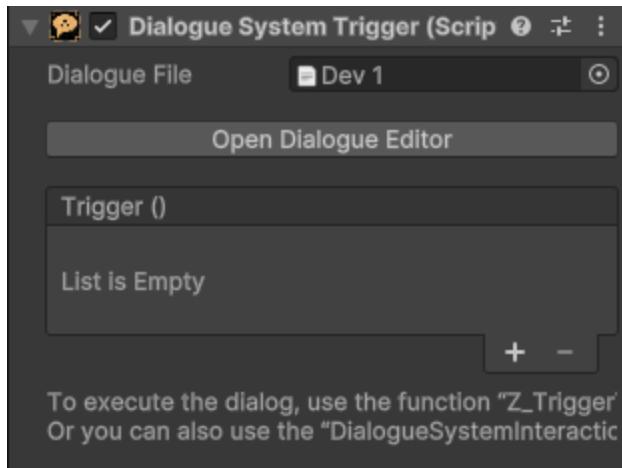
- Time show Letters: Refers to the playback time between letters
- Time show Comma: Refers to the playback time of commas
- Time show point: Refers to the playback time of the points
- Every How many letters sound: refers to every how many letters the speaking sound is reproduced

GUI

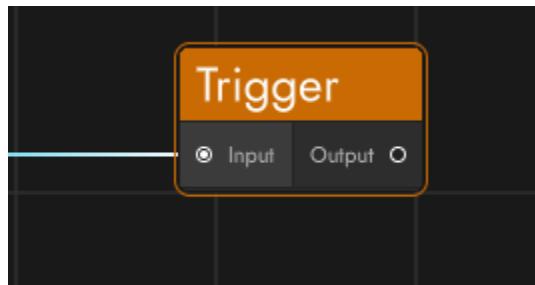
- Panel: This refers to the entire dialog panel, it is where all the elements to make the dialog are. This is necessary to function.
- Panel text: This is the dialog panel
- switch panel: This is the panel where the options and modifications to the conversation will appear.
- switch button: This serves as a model for the buttons on the switch panel.
- dialogue text: The Text mesh of the text box.
- bg dialogue text: The background of the text box.
- Character right: All character elements displayed on the right of the screen appear here
 - name panel: It is the panel where the character's name appears
 - character name text: The Text mesh of the character name.
 - character animation: The character animation, this is used to execute animations,
IMPORTANT: it works with the Animation, not with the animator.
- character left: the same as "Character right" but with the character that appears on the left.
 - ...
- Events: these are events that are executed when certain things happen, very useful for implementing functions without having to touch the code.
 - Dialogue
 - Start Dialogue: when a new dialogue is started
 - End Dialogue When a dialogue ends
 - character's turn:
 - Right Character Turn: When the character on the right speaks
 - Left Character Turn: When the character on the left speaks
 - Switch
 - Turn On Switch: This appears when the Switch panel is displayed

Dialogue System Trigger

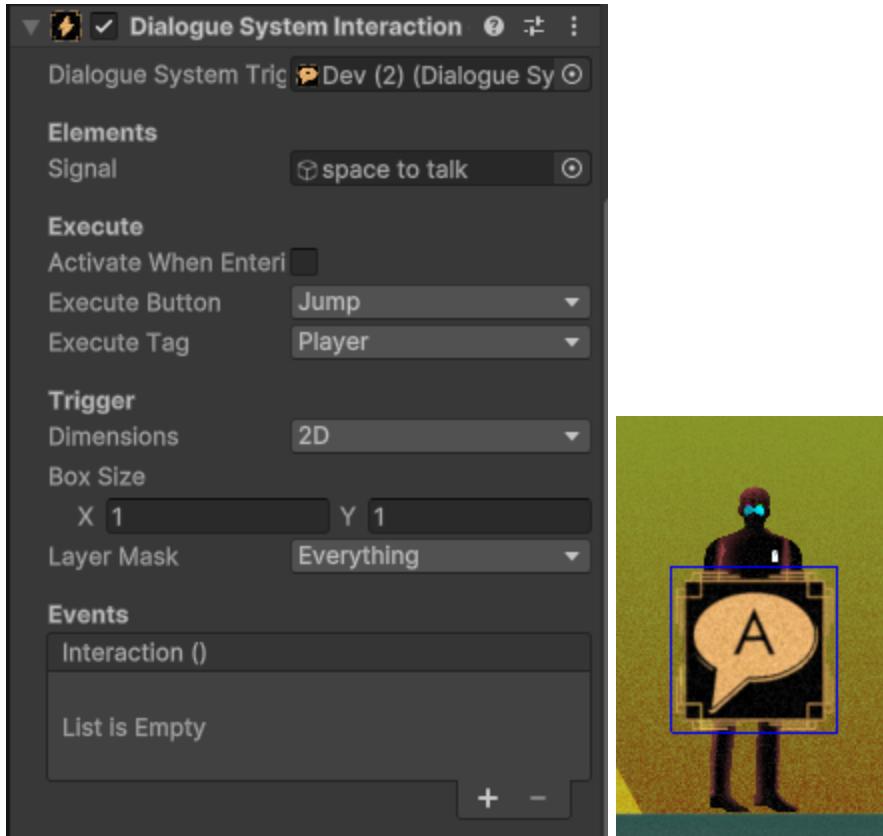
This component is responsible for executing the dialogue.



- Dialogue file: The Dialogue file that will be executed appears here
- The button:
 - Create new dialogue file: If you do not have any dialogue file in place, the option to create a new one appears here.
 - Open dialogue editor: This opens the editor to edit the dialogue file.
- Trigger:
 - This is executed from the dialog with the “trigger” function within the Dialogue Editor Graph



Dialogue System Interaction



This executes the dialogue trigger when the player enters the blue collision box and a specific button is pressed.

As each project is different, we recommend using your own system that executes the “Z_TriggerTheDialogue()” function that is inside the Dialogue Trigger

Elements

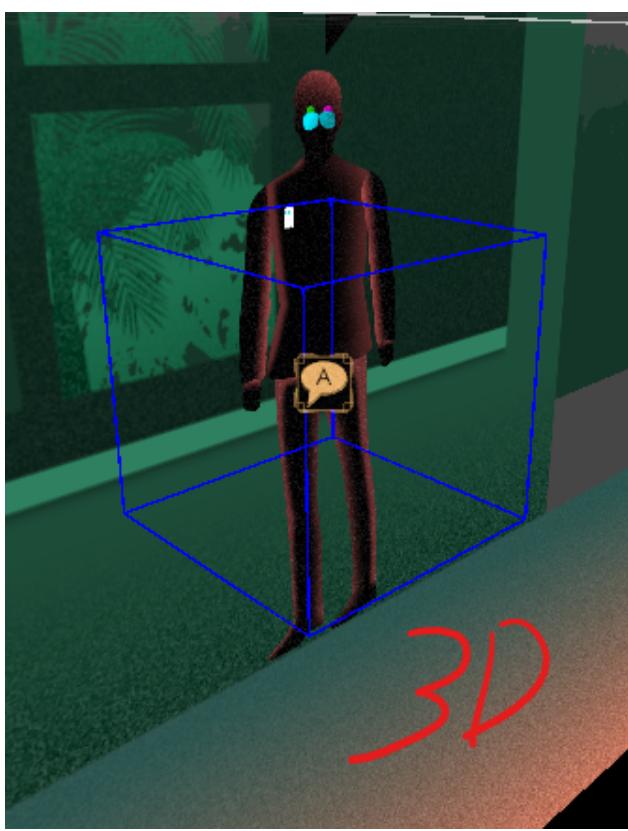
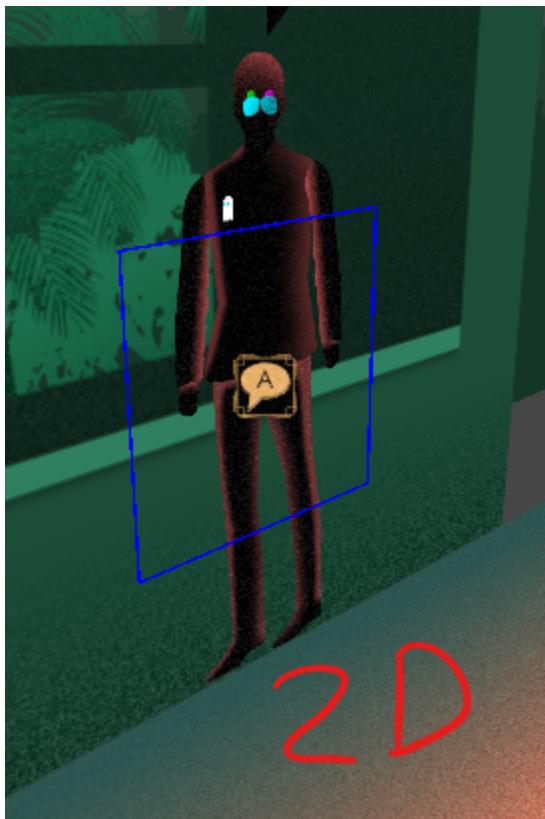
- Signal: When the player enters the blue collision box, this GameObject is activated.

Execute

- Active when entering: If activated, it will be executed when the player enters the blue collision box without needing to press anything.
- Execute button: Here you choose the button that executes the dialog. This uses the old input system and refers to the values per effect.
- execute tag: Here, you choose the tag that will execute the dialog.

Trigger

- dimensions: Here you can switch between a 2D or 3D project

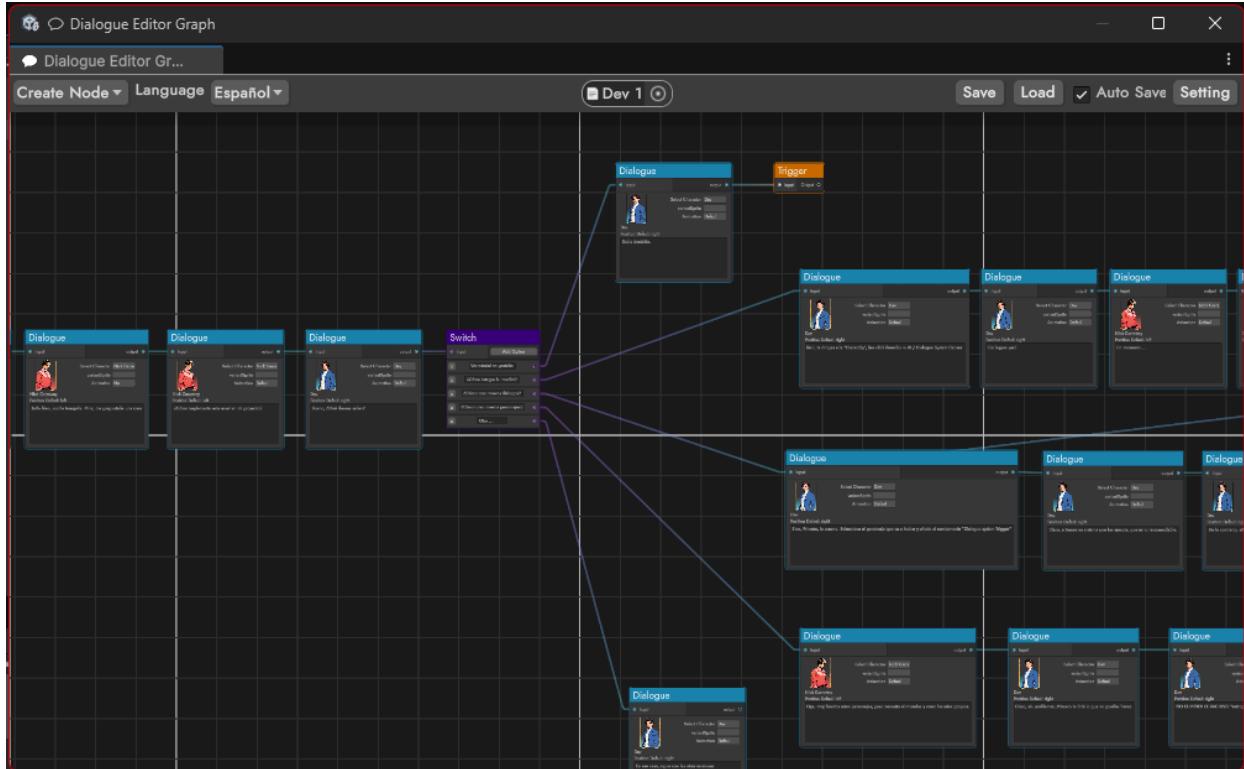


- box size: Dimensions of the four blue collision box
- layer mask: blue collision box layer

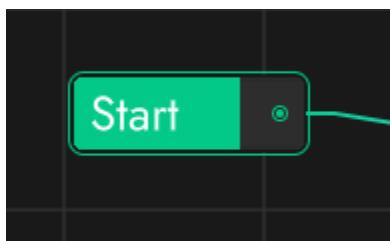
Events

- Interaction: when the dialog is executed

Dialogue editor Graph

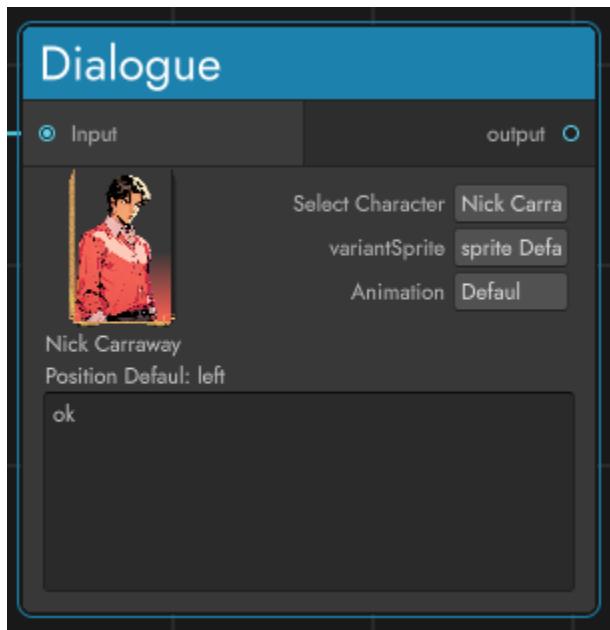


- Create node: in this panel you can create the nodes that will make up the conversation.
 - Start: With this node the conversation starts. If you put more than one start, when starting a dialogue, it will choose one at random.

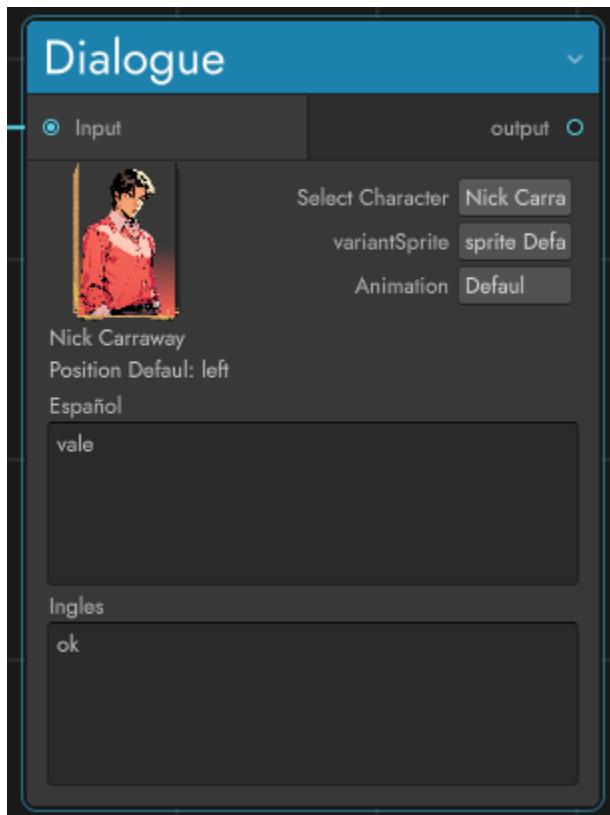


- Dialogue: This is where the characters' dialogues are written:
 - Select Character: Select the character who will speak
 - VariantSprite: we select the variation of the sprite.

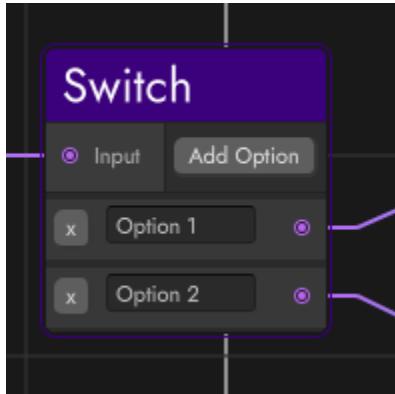
- Animation: We select the animation when speaking.



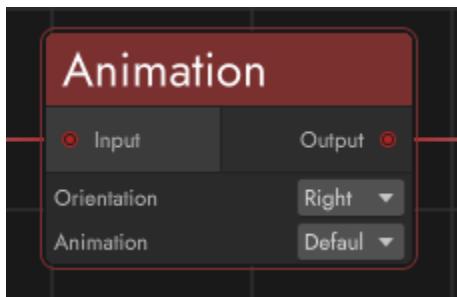
If you select “Show All” all the languages you have created will be shown.



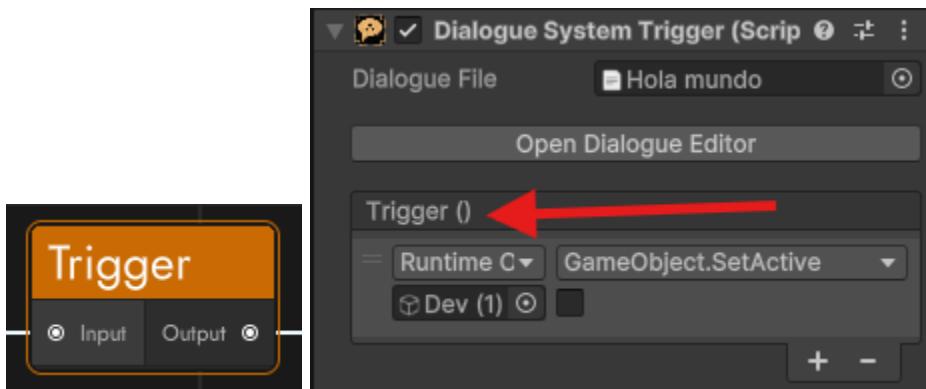
- Switch: With this you can make decisions that will affect the narrative of the project.



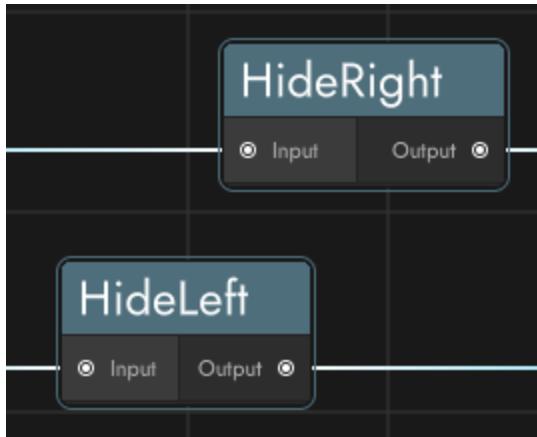
- Animation: Runs an animation without dialogue.
 - Orientation: refers to the character who will do the animation, the character on the right, or the one on the left.
 - Animation: The animation that will be



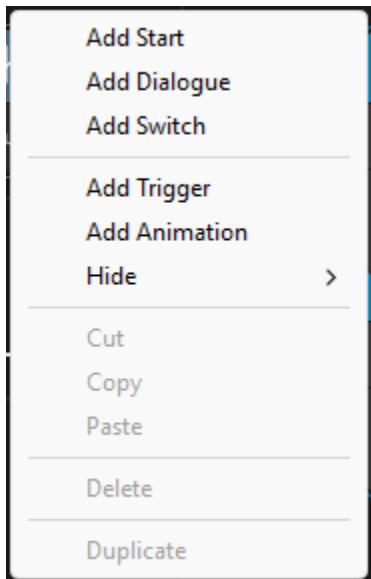
- Trigger: This executes the trigger event of the Dialogue Trigger



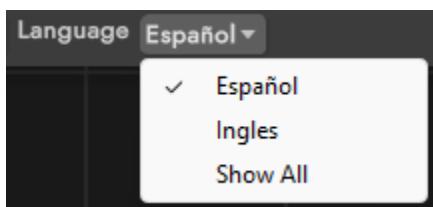
- Hide: Hides a character, either the one on the right or the one on the left.



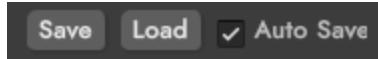
- Usability features:
 - With right click you can create the nodes, these will be created at the current mouse position.



- Language: we select the language that the editor will display. In “Show All” all languages will be shown in the editor, Ideal for making translations.



- Save, Load and auto save: Pretty self-explanatory, Save, Load and Auto Save. The auto Save is still in development, I recommend clicking save anyway.



- Setting: This comes to you at Dialogue System Setting



- dialogue file: This is the dialogue file that is currently being edited. We will soon add features to create Dialogue Files directly from the editor.

