Thank you for purchasing Geppetto, the animation tool for lip-sync and facial animation.

This document will show you how to use the asset in Unity (version 2021 or later).

If you have any question, contact us at contact@xandimmersion.com or on discord https://discord.gg/qDMwNCDE8X

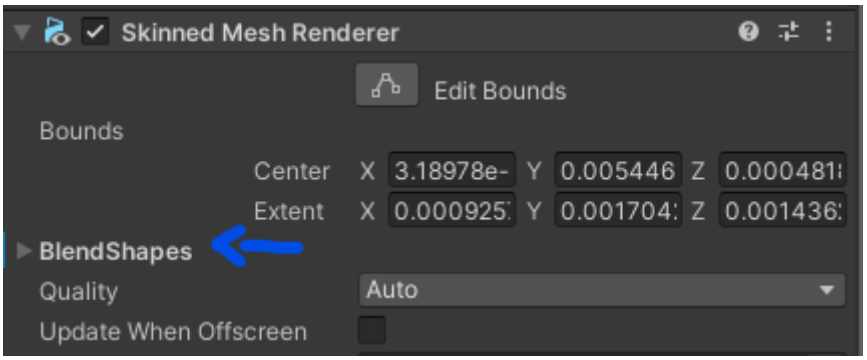
0 - Introduction about blendshapes and phonemes

⚠️Please take note that our animation tools **only work with Blendshapes** so make sure your character already has some. If not, you can use software like blender or maya to create them.
⚠️ In order for you to generate the animation you must have an **internet connection** as it uses our **internal API** for the generation. Once your animation is generated, you don't need to be online to be able to use it.

Définition:
Blendshape is a deformation of a mesh to achieve a predefined pose.
Here, the blendshape that we use for our facial animation are blendshape reproducing a phoneme.
Phonemes are the smallest units of sound in a language and are used to differentiate one word from another. For example, the words "bit" and "bet" differ by only one phoneme, the "i" sound in "bit" and the "e" sound in "bet".
A **blendshape of a phoneme** is a blending from a closed mouth to a mouth pronouncing the desired sound.

Our technology uses up to a total of **39** phonemes that we adapt based on the 3D model's blendshapes.
You can easily create your blendshapes on your 3D model in Blender, Maya or any 3D model software that allows you to work on your mesh.

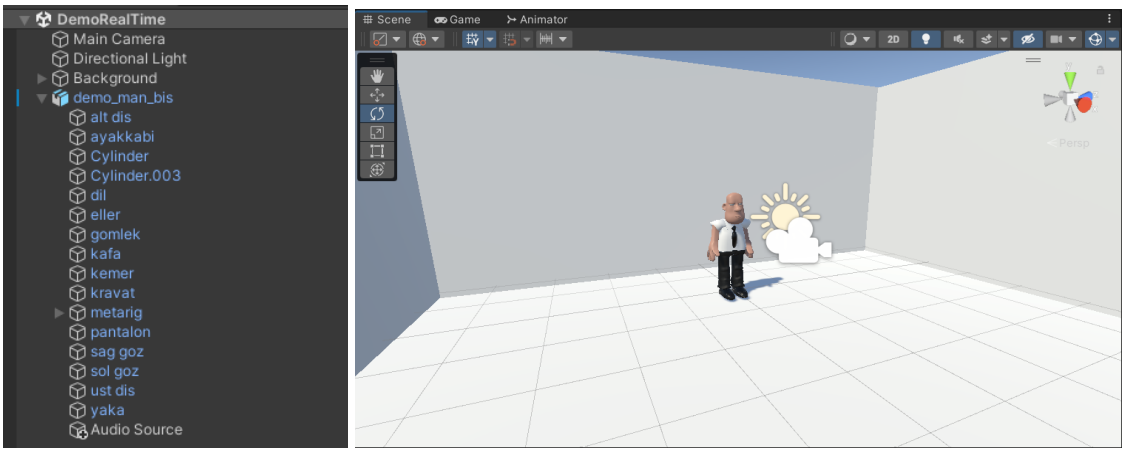
When importing your 3D model on the scene, you can find your blendshapes under the SkinMeshRenderer component like so.



You can also check if you have blendshapes directly in Blender/Maya or other 3D modeling software.

1- Learn how to use the tool with the demo scene

Hierarchy **Demo**



⚠️**In order to have the Demo Scene you must first download it from out GitHub repository :** X-Immersion/Geppetto_demo (github.com)

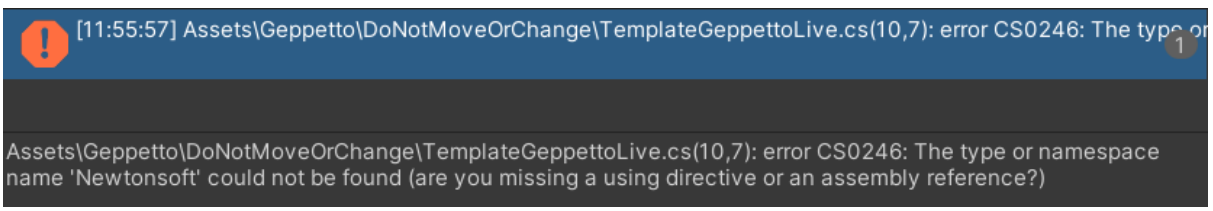
You will find under Geppetto/Exemple/Scene, 2 scenes for you to try our tools, **Demo** and **DemoRealTime**.

There are two ways to use this Geppetto tool. Either you want to use it to save production time and **pre-generate** all your lip animations from existing texts or audios. In this case, look at **section 1**.

If you want to run Geppetto dynamically in your game and generate the animation in **real time**, see **section 2**. You may need speech synthesis or text generation for your characters. You can look at these tools on the unity asset store if you are interested.

Do not hesitate to contact us for further information at contact@xandimmersion.com

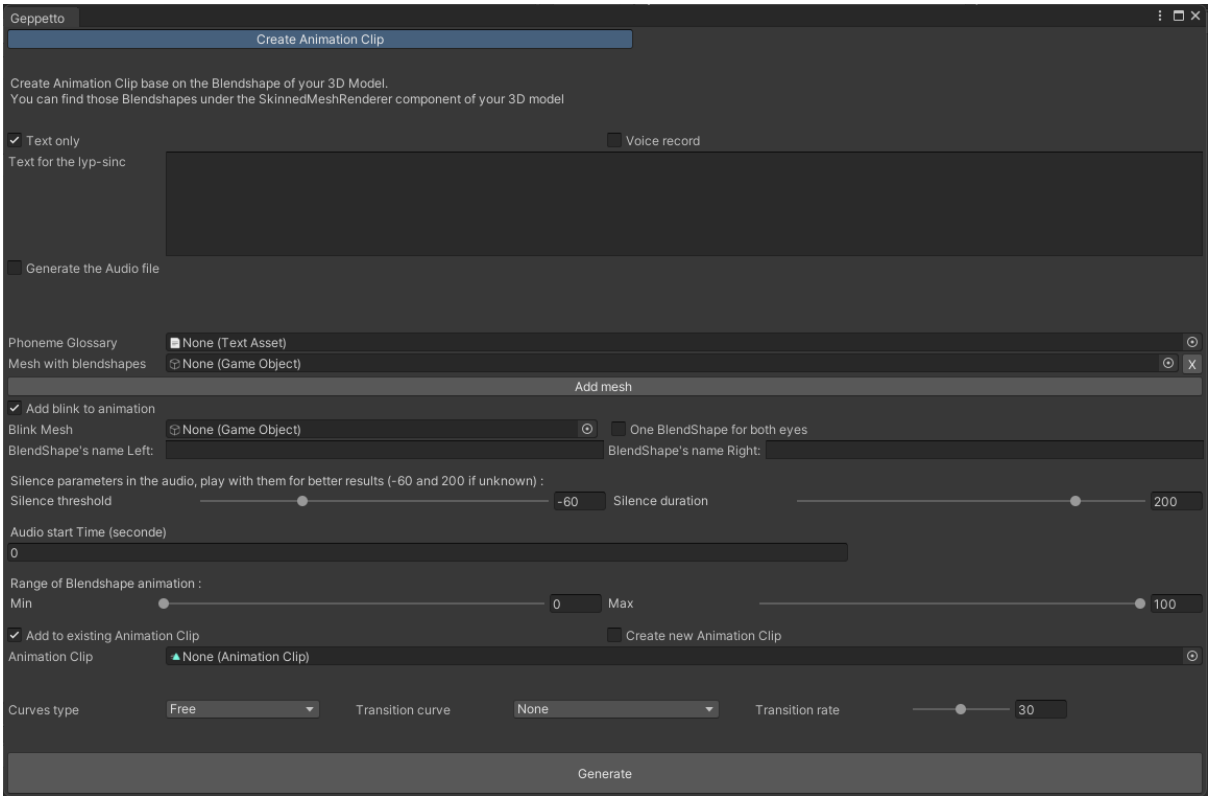
⚠️If you're experiencing the following error : link



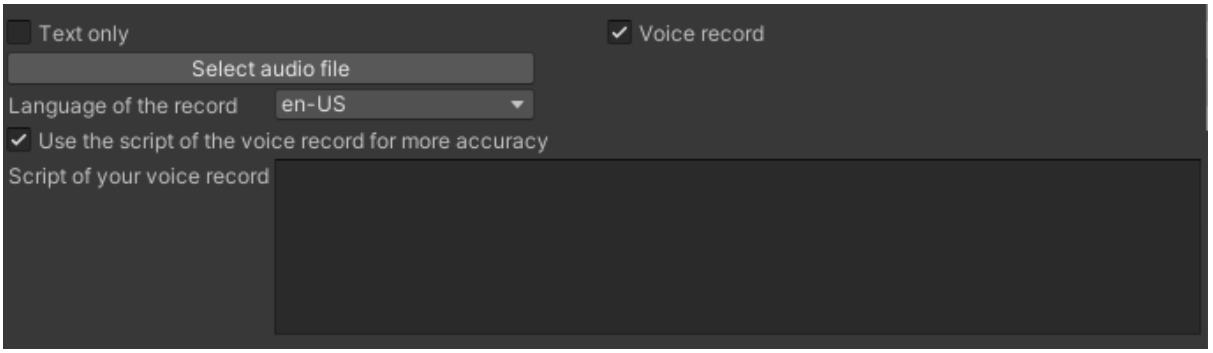
It means you don't have the required version of Newtonsoft json or that it is not correctly set up in your project. Follow this procedure: [Json.NET - Newtonsoft]

1a- Pre-generate the animation clips

In the Editor, open Window/Geppetto.



If you have an audio file, click on **voice record.** The animation will be synchronized with the text.



⚠️ Please note that for every pause in your audio file, you need a comma " . " at the corresponding place. As we need those in our algorithm for precision purposes.

⚠️If you have vocalization in your audio, for example *grunts*, try using simple onomatopoeia like “uh” on your text instead. As we do not support tags yet.

1b-Add the Phoneme Glossary

You can skip this part if you already have one

1. Duplicate then Open the spreadsheet named “glossaryFaceIt.csv” (Assets/Geppetto/Example) in your preferred software application, such as Microsoft Excel or Google Sheets.

   You will see 2 columns:
   - The first one is the "Phoneme" column. It contains a list of the 39 phonemes that we use to animate 3D characters.
   - The second is the "Blendshape'' column.

2. In the first row of the "Blendshape" column, enter the blendshape that corresponds to the first phoneme listed in the first column. For example, for “P", you may enter the blendshape that makes your character's lips close tightly to simulate the "p" sound.

3. Repeat step 2 for each subsequent row, entering the blendshape that corresponds to the phoneme listed in the same row of the first column. Make sure to enter the correct blendshape’s name.

4. Save the file as a CSV (comma-separated values) file, which is a standard format for storing spreadsheet data.

5. Test the file by opening it in a text editor to ensure that each row contains two values separated by a comma (one for the phoneme and one for the blendshape). You can also test the file by importing it into your 3D modeling software and applying the blendshapes to your character's facial rig to ensure that each blendshape corresponds correctly to its corresponding phoneme.

6. If necessary, make any additional changes or updates to the file and repeat steps 4 and 5 until you have achieved the desired results.

Great job! You've completed the most challenging part of this tutorial.
You now have a CSV file with 39 phonemes in the first column and their corresponding blendshapes in the second column, which will be used to control your 3D character's facial expressions when animating speech.

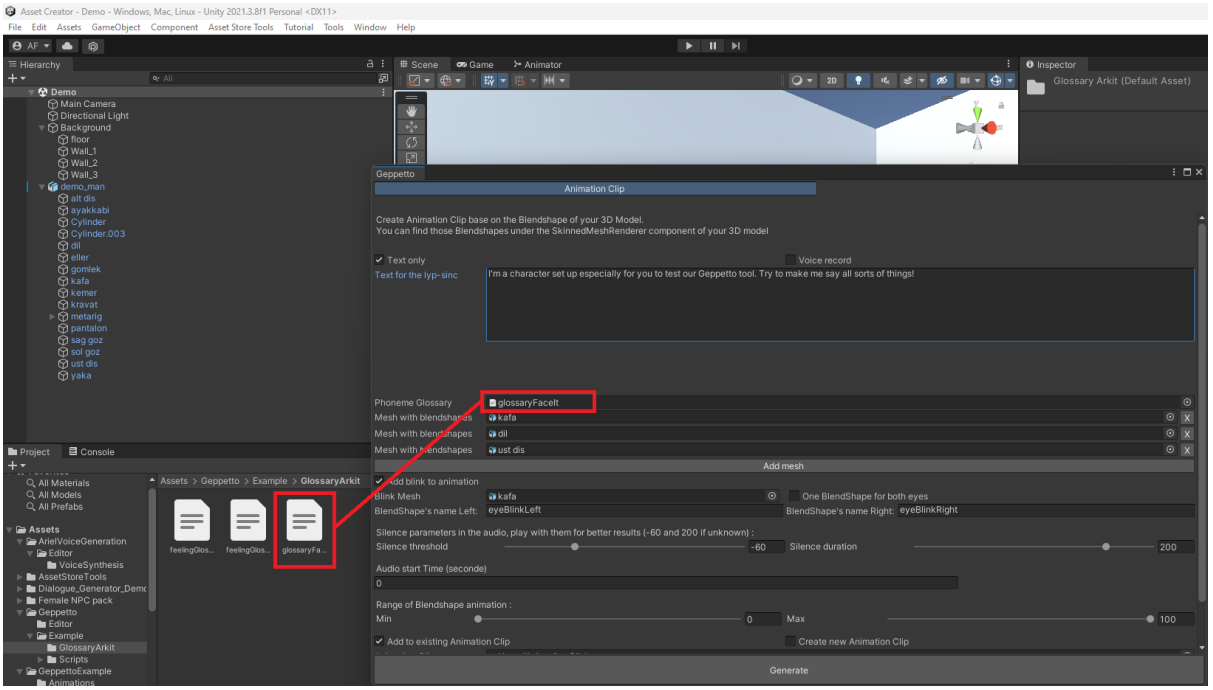
Example : How you would fill it for your Character

AA	myBlendshapeFor_AA
AE	myBlendshapeFor_AE
AH	myBlendshapeFor_AH
AO	myBlendshapeFor_AO
AW	myBlendshapeFor_AW
AY	myBlendshapeFor_AY

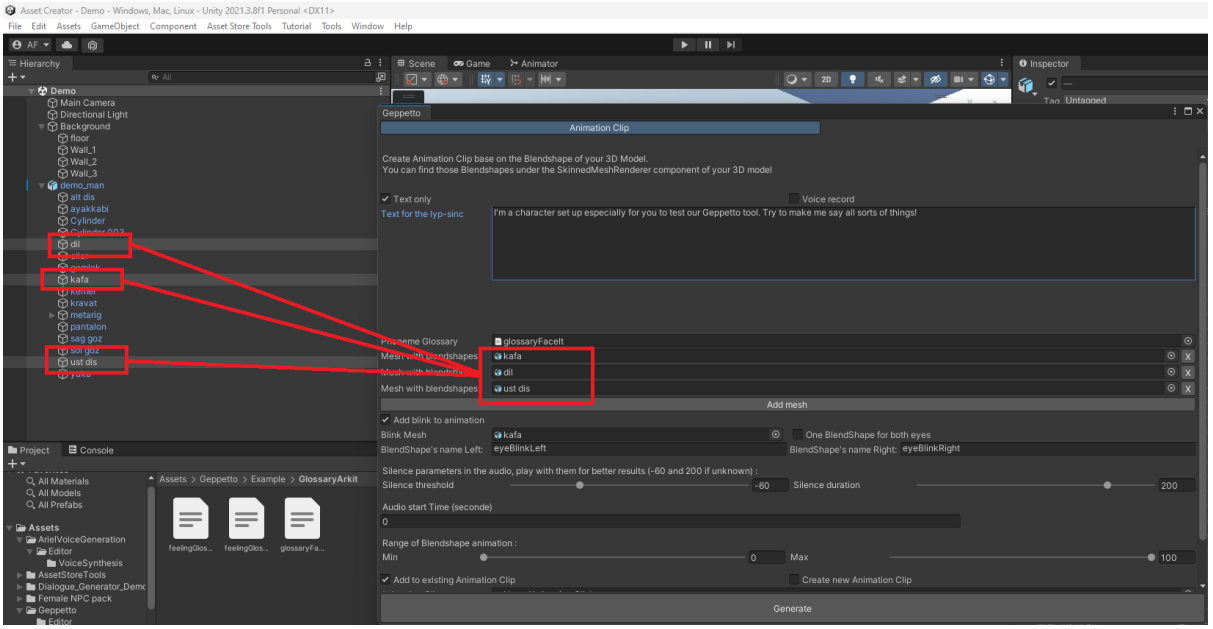
You can use the same Blendshape for multiple phonemes

15	G	myBlendshape_GK
16	HH	myBlendshape_I
17	IH	myBlendshape_I
18	IY	myBlendshape_I
19	JH	myBlendshape_I
20	K	myBlendshape_GK

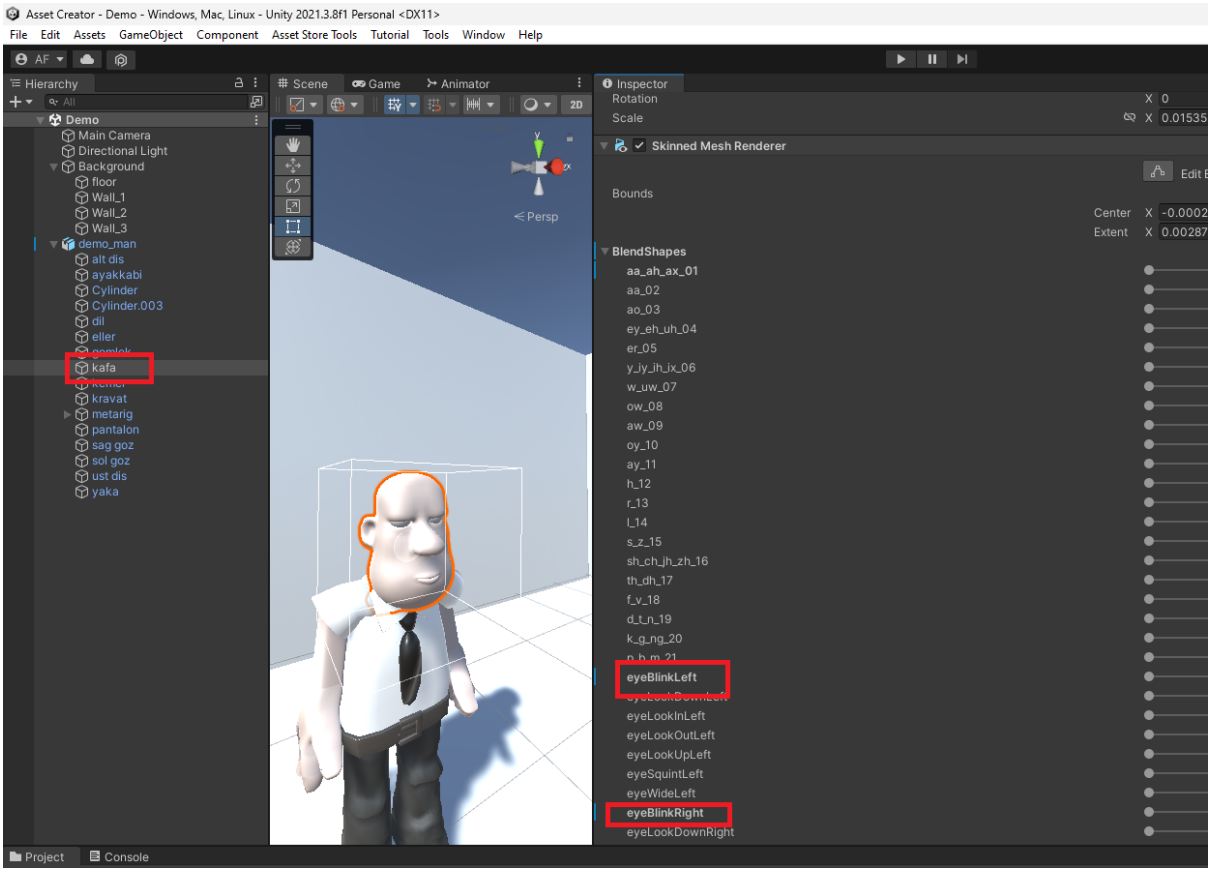
1c - Locate your csv file and drag it into the Geppetto editor



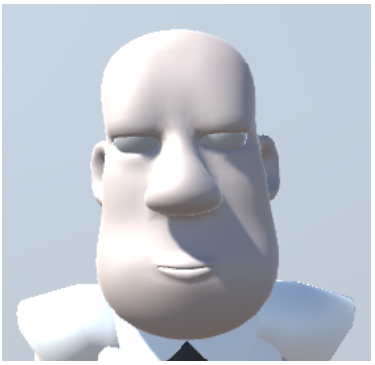
1d - Find your object with the SkinnedMeshRenderer attached to it



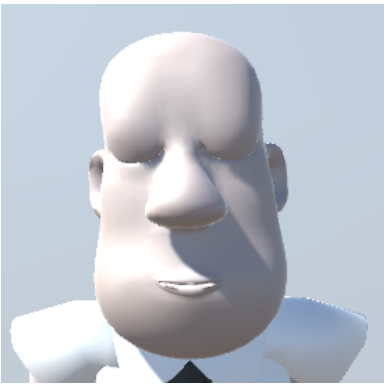
1e - Add the blinking



eyesBlinkLeft/Right = 0



eyesBlinkLeft/Right = 100



1f - Silence parameters

Those parameters will vary depending on the voice and recorder you're using.
Silence threshold is referring to the maximum intensity present in your audio for the noise, in dB.
Silence duration is referring to the minimal duration of silence in your audio.
They are used to differentiate what is voice and what is a silence in your audio file.
If you're using only text:

- silence threshold = -60
- silence duration = 200

1g - AudioStart Time

This parameter is to set the right timing for the beginning of your audio in the animation.
For example, if your animation is 2 minutes long and your audio is 30 seconds and should start at 43s.
Or if you're working with multiple audio.

1h - Curves Type

Unity animation curves can be classified into five different types:

Free: Free curves are the most flexible type of curve. You can manually add and position keyframes to create any shape of curve you want.

Auto: Auto curves are automatically generated by Unity based on the keyframes you have placed. Auto curves are typically smoother than free curves, but they are less precise.

Linear: Linear curves are the simplest type of curve. They change value at a constant rate over time.

Constant: Constant curves have a single value that remains unchanged over time.

Clamped Auto: Clamped auto curves are similar to auto curves, but they are clamped to the values of the first and last keyframes. This means that the value of the curve cannot go outside of the range defined by the first and last keyframes.

1i - Transition Function

A transition function, also known as an easing function, is a mathematical curve that defines how a value changes over time. Transition functions are used to create smooth, animated effects in Unity, such as moving a game object from one point to another, or changing the color of a UI element over time.

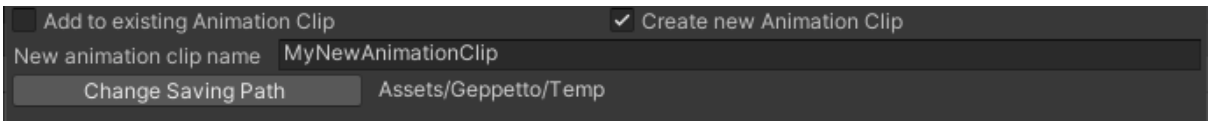
There are many different types of transition functions, each with its own unique properties. Some common transition functions include:
Linear: The value changes at a constant rate over time.
SmoothStep: The value changes at a gradually increasing rate, and then gradually decreasing rate.
EaseIn: The value starts slowly and then accelerates over time.
EaseOut: The value starts quickly and then slows down over time.
EaseInOut: The value starts slowly, accelerates, and then slows down again.

1j - Transition Rate

Number of keyFrame for every curve.

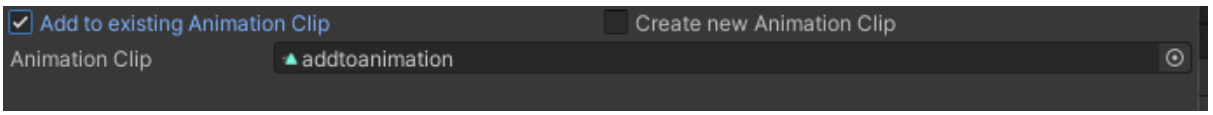
The higher the number, the more accurate your transition function will be.

1k - Create new AnimationClip



1l - Add your blendshape animation to existing AnimationClip

If you want to add lip sync to an existing animation, select the animation here.

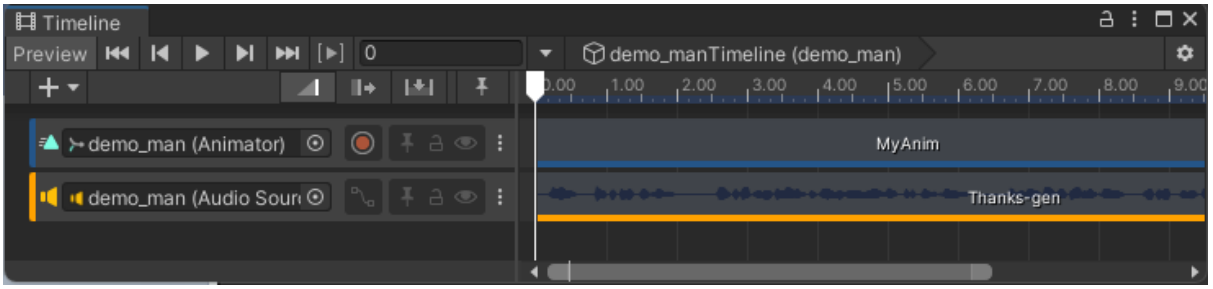


1n - Try it out with a Timeline

When everything is generated, you can create a Timeline (Window > Sequencing > Timeline).

Create an AnimationTrack, drag and drop your character in it, drag your AnimationClip. (You may need to add the AnimationClip to your character AnimatorController beforehand)

Create an AudioTrack, drag and drop your object containing the AudioSource, drag and drop your AudioClip



You may need to run your project one time for the timeline to update.

That's it !

Test it in Runtime

2a - Use the run time script

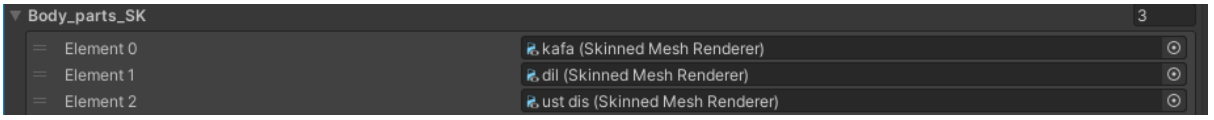
1. Go to the Unity scene where you want to attach the script to a GameObject.

2. Select the Parent GameObject that you want to attach the script to.

3. In the Inspector window, click on the "Add Component" button and search for "ParametersController".

4. Click on the "ParametersController" script to attach it to your selected GameObject.

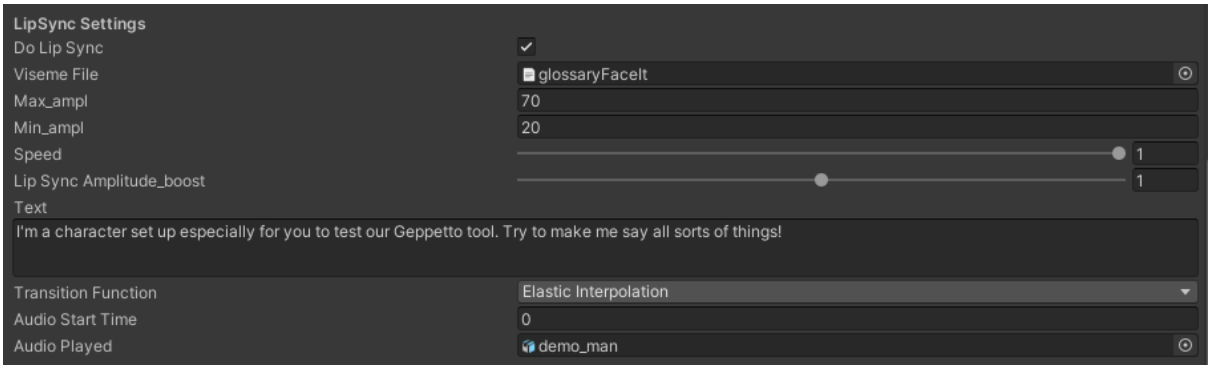
5. Drag and drop every mesh object containing shape keys into the “Body_part_SK”



6. Drag and drop your glossary file into the "Phoneme Glossary" bar in the “VisemeFile”

7. Select the range/speed of the animation. The Amplitude_boost parameter is to multiply the amplitude you get from the generation

8. Fill in the text you want for the generation



Your script is now attached to your GameObject and you can start filling it to control your character's facial expressions based on the phonemes in your glossary.