**Things to Learn:**

**Scope**

**Objectives:**

**Promote to:**

**Research**

**Tags**

free lipsync

free lipsync unity

**Credits**

* Radish Modding Tools: <https://www.nexusmods.com/witcher3/mods/3620?tab=description>
* Witcher 3 Phoneme Extractor Source Code: You will need to build this first <https://codeberg.org/rmemr/w3.phoneme-extractor>
* Links: https://github.com/FFaUniHan/Radish\_Tools\_Phoneme\_Extractor

# Things to Do

# Setup

**0.1**

What you’re seeing here is a recorded lip sync animation running in Unity. This plugin doesn’t use any video files or webcam. This is done by inputting the recorded audio file into a program, and then passing the result through a plugin that makes a Unity animation file for the mouth movement.

*Subtitles are available in English!*

**0.2**

This method works for any models that have mouth movement blendshapes for AEIOU.

*In this example, I’m using a VRoid model.*

**0.3**

As far as I know, I didn’t find any method this easy and free while searching through youtube and Unity subreddits. So here’s my attempt at it.

# 1. Recording Voices

To get started, you will need to record some voice lines. Make sure your audio file is a **16-bit .wav format**, **mono channel, 44.1 khz**. If your voice lines are longer than 15 seconds, cut it into several files. I’m pretty sure you can find how to do this online, but do let me know if you want me to create a tutorial for recording audio and saving it in the right format.

# 2. Downloading Radish Tools

**2.1**

The program we’re going to use to generate lipsync is actually a community modding tool for Witcher 3 called Radish Tools. It is a set of programs that is used to create custom quests, maps, and as we are interested today, phoneme extractor for lipsync.

You can download this entire tool in NexusMods, which requires you to login or create an account. Or you can download this from my github page, link is in the description.

***2.2***

*It uses the GNU GPLv3 license, which should mean that everyone can use it commercially for free as long as they didn’t publish the tool itself. Remember that this is not legal advice, though*

***2.3***

*I also asked the creator directly on Discord, and they’re okay with us using it for indie game developments. Hurray!*

# 3. Using Radish Tools

**3.1**

In the Radish Tools folder, the first thing we need to do is to put all of our audio files into the **audio.wav** folder. We need to rename them into a series of 10 digit numbers.

**3.2**

Then, we want to open **template.speech.csv** file in Notepad. Here, we have to type in all the file names in the **audio.wav folder** earlier, along with the lines of dialogue inside each file. Make sure there are no empty lines at the end of the file. Once you’re done, save this file and close it.

**3.3**

Now, we want to run the Phoneme Extractor exe file and here is where the magic happens. Click on the **File** **menu** and choose **Load Audio**. We want to locate our **audio.wav** folder and click **OK**. And now we wait. If there is no error, you will see the process running on the bottom of the screen. Once everything is done, we can close this program.

**3.4**

If we open our **audio.wav** folder again, you can see that we now have a pair of **wav** and **phoneme** files. We will use these phoneme files as the input in the next process.

But what is a phoneme? I will explain it later in the video.

# 4. Downloading UniLipSync

For the next step, we want to download a Unity project file from my Ko-Fi page.

*Don’t worry it’s free, but I do appreciate some spare change here and there.*

Then, open this project in Unity.

# 5. Using UniLipSync

**5.1**

First of all, move your audio files into the **Voice Over** folder in our Unity project

**5.2**

To create your lipsync animation, in the **Hierarchy Window**, click on the **Generate Lipsync** game object and click on this button. Then select your **audio.wav** folder from the Radish Tools earlier.

You can see your lip sync animation files generated in the lipsync folder, with the same ID name as your audio and phoneme files.

If you tick the **Auto Add to Animator** option and fill in the **Animator** field, you should see that the lip sync animation is added automatically into the animator.

**5.3**

To test the lipsync animation, we can go into **Play Mode**. In the Hierarchy Window we can open the **Audio Tester** gameobject. Drag in the audio files one by one and click on the button to play it.

**5.4**

We can also put these animations and the audio into Timelines and create a cutscene for it.

# 6. Customization: Smoothing

Optionally, before clicking the button, we can fine-tune our lipsync animation with these four properties, all in seconds.

[Video shows hovering over Attack and Releas Time]

**Attack and Release Time** determines how fast your character opens and closes their mouth. For most cases, you don’t need to change these values.

**Hold Time** determines how long the character should keep their mouth open between phonemes. So, if there are a lot of pauses in your voiceline, you may need to lower this value.

**Smoothing Time** helps if you find a lot of mouth movement to be too quick when the character speaks several phonemes in rapid succession. Raising this value will combine multiple phonemes into a smoother animation. Of course, if you raise it too high, the lip sync animation will start to look inaccurate or kind of delayed.

# 7. Customization: Viseme Group

And then if you find some mouth movements that don't quite match the voice, you can open this ScriptableObject called **Viseme Groups**.

Inside this ScriptableObject, there are 9 items each containing a list and some sliders. This list contains some weird characters called **phonemes**. You’ve seen this in a dictionary. Each of these symbols represent a single sound humans can make with our mouth.

[With example: Cat]

When spoken, sometimes your mouth will make a similar movement for different sounds, like /b/ and /p/, or /d/ and /t/. Try it. We call these Visemes, or different sounds that have similar mouth shapes. And these groups in the ScriptableObject are those Viseme Groups. All of the phonemes in a group will make similar mouth shapes. And the sliders for the vowels is my best attempt to match the mouth shape of each group.

So, if you feel that some of the sounds don’t quite match the mouth animation, you can easily move some phonemes to another group or create your own group entirely.

# 8. More Blendshapes

Of course, with just five blendshapes, we are very limited in matching real life human mouth shapes. In the future version of this plugin, I would like to achieve better lip sync with the 52 blendshapes AR kit.

*A more extreme way to customize this plugin is when you have different blendshapes for the mouth that are more than just vowels. For example, if your model uses the 52 blendshapes AR Kit, you need to add and replace quite a lot of parts for the AEIOU in the script and replace it with your own blendshapes. It could take a bit of time, but the rest of the code should work fine. Do let me know if you need help in the comment section below.*

# Closing

If you like this video, you can support me on Patreon like Javi here, where you can watch videos a week earlier and test out new plugins I made before anyone else. Like and subscribe if you want to see more videos from me. And with that, I think I’ll see you guys later, goodbye!

# 6. Understanding Linguistics

But if you want to do more fine tuning, then we need to understand some simple linguistics and disney animators. Let me explain.

**6.1**

So, let’s start at the names of our files earlier. Phonemes are sounds we make when we talk. They are like building blocks for words. For example, “cat” has three phonemes: /k/, /æ/, and /t/.

The problem is, English is such a messy language and different words can sound the same, like *sea* /siː/ (a large body of water) and *see* /siː/ (to perceive with eyes). And different sounds can be written the same, like *l*[*ead* /lɛd/ (a type of metal) and *lead* /lid/ (to guide someone)](https://www.speechactive.com/english-homophones-list-examples-pronunciation/).

**6.2**

So to make things easier to learn, everyone agrees to use this type of notation so everyone knows exactly what sound we’re supposed to make. This is called the International Phonetic Alphabet or IPA and it is *a lot*… In fact, this entire table contains all sounds that a human mouth can make.

So to save us from the headache, from now on, we’re going to focus solely on sounds we make in English. To be more specific, American English.

[tomato: /təˈmeɪtoʊ/](https://www.angmohdan.com/22-words-with-british-and-american-pronunciations-that-may-confuse-you/) (American) [vs /təˈmɑːtəʊ/](https://www.angmohdan.com/22-words-with-british-and-american-pronunciations-that-may-confuse-you/) (British)

garage: /gəˈrɑʒ/ (American) vs /ˈgærɑːʒ/ (British)

yogurt: /ˈjoʊgərt/ (American) vs /ˈjɒgət/ (British)

zebra: /ˈziːbrə/ (American) vs /ˈzɛbrə/ (British)

But you can easily apply this to any language you want.

The English language has 44 phonemes, which you can read yourself on the screen right now.

*[Show phoneme tables in ARPANET for Radish Tool]*

*Let’s focus on the vowels first. If you try to pronounce the vowels from left to right, you will notice that your tongue moves further to the back or your mouth. And as you move down the table, your mouth opens wider. That’s how they organized this table.*

*The consonant table is a little bit more complicated. The columns are still where your tongue is in your mouth but it is more specific, like at the place where you burn your mouth when biting a hot pizza (alveolar). The rows are how you’re supposed to articulate it, like a single burst (plosives) or from your nose (nasal).*

**6.3**

So what’s the point of me explaining this? The point is that in speaking, our mouth needs to move in a lot of different ways. And this is a massive nightmare to draw if you’re drawing frame by frame, by hand… Which is a problem that Disney animators faced in the 1920s.

Remember that there were no animation programs or even computers back then. Fortunately, they found a shortcut. They noticed that mouth movements can look similar even if their sounds are different, like /p/ and /b/ or /d/ and /t/. This is called **visemes**, or different sounds that have similar mouth shapes. And we can easily put these phonemes into groups, or **Viseme Groups**.

So now, instead of drawing 44 mouth shapes, they can just draw 12.

*And this is still great news for digital animators today, because we don’t need to make too many Blendshapes in our 3D model.*

**6.4**

For us, this means that we only need five vowel blendshapes in different values for each one of those Viseme Groups.

So here are the 9 Viseme groups I came up with. I put this into a ScriptableObject containing the list of the phonemes and the blendshape values for the 5 vowels to make the mouth look like the visemes.

## Limitations

There are several limitations

* The Witcher 3 Phoneme Extractor plugin is slow. If you guys have any experience in audio linguistic analysis, let me know.
* Tongue movement through Blendshape

## Further Improvements

Here are the list of improvements

* The plugin can recognize the levels of voice: High, Medium, Low. High means that the user is shouting, Medium is a normal talking voice, Low is when the user is whispering. This can be done easily through analyzing the amplitude of the sound clip.
* Facial stress contortion. The plugin can influence how the character’s face will change when emphasizing a word, like a slight movement of eyebrows, look at Cyberpunk 2077 plugin for lip sync.
* Making this tool work for all languages
* Adding more tongue movements