

# Chatterware - 'Now We're Talking!' - v1.07

Giving characters a voice, so to speak!

Welcome to 'Now We're Talking!' (NWT). Thank you for buying our asset.

## Getting to know 'Now We're Talking!'

The best way to get to know how 'Now We're Talking!' works is to see it in action and watch it working. For this we have provided a demo and a number of tutorials you can find on the website. <https://www.chatterware.com/tutorials>

## Using the Demo Scene:

We assume you have already downloaded and imported 'Now We're Talking!'. Once you have imported the 'Now We're Talking!' package into Unity, locate the folder structure in your Unity project, in the Unity editor.

Load the "NWT\_Demo\_001" scene from the "Scenes" folder in the "Now We're Talking" folder.

Look at the hierarchy of the scene in the Unity Editor. Locate the 'Now We're Talking!' prefab in the Hierarchy. It can be found as a child of the 'Sphere' object.

Inspect the 'Now We're Talking' gameobject and you will see a deactivated child object called 'NWT\_PlayAudio'. In the Game Window deselect 'Maximise On Play' because you will need access to the 'NWT\_PlayAudio' gameobject to play the voice clips when you run the scene.

Run the scene. Make sure you can comfortably see the 'mock up' face, the Hierarchy tab and the Inspector tab. You will see a 'mocked up' character face made up from some simple Unity primitives. This scene represents the basic elements 'Now We're Talking!' can control on a character, using a voice clip. We have already set up the scene and pre-loaded voice clips so users can quickly see what 'Now We're Talking!' is capable of.

With the scene running, activate and deactivate the 'NWT\_PlayAudio' gameobject. You can use the mouse to enable and disable the gameobject by clicking in the checkbox, in the 'NWT\_PlayAudio' gameobject in the Inspector tab, when you want the 'character' to speak.

The 'mouth' should now start animating in sync with the voice clip which plays. When a voice clip is finished playing, disable and enable the 'NWT\_PlayAudio' gameobject in the inspector at any time, to play another voice clip.

## You should see is the following:

- (1) an animating mouth, moving in sync with the voice clip, playing
- (2) the two objects which represent 'eyes' scaling up and down
- (3) the two objects representing 'eyebrows' scaling up and down
- (4) the two capsule objects and the squashed cube object representing meshes which can flash and/or rotate in the case of a vehicle or science fiction robot or mechanical character.

Adjusting these parameters allows your many different NPC's (none player characters) to have many different mouths.

Whilst the scene is still running, select the 'NowWe'reTalking' gameobject in the Hierarchy. The Inspector tab should now show the Component (script) 'Now We're Talking'. Look carefully at the many parameters which are available to set in this asset. We have provided some 'work out of the box' data presets for you with a number of voice clips. They can be found using the 'Mouth Data Presets' slider. Locate this slider.

Try moving this slide from position 1 to position 2. Now try playing another voice clip, enable and disable the 'NWT\_PlayAudio' gameobject as described above. You will see a rectangular shaped mouth, useful for sci-fi, robot and mechanical characters, maybe even autonomous vehicles. Now change the 'Mouth Data Presets' slider to position 3 and play another voice clip. Look at the Unity Inspector and locate the NPC\_Phases section of the Now We're Talking! Script. Here you will see the lists of voice clips which operate the mouth shapes. See (1)\* below for full details.

Congratulations you have completed the demo scene. Now is a perfect time to read the documentation and watch our short, fun video tutorials on the Chatterware YouTube channel. Good luck and we hope you enjoy using 'Now We're Talking!'.

Please read the following documentation, follow the many video tutorials on our website or YouTube channel and visit our Discord channel to get a full explanation of the **Now We're Talking!** asset and it's huge potential.

'NowWe'reTalking' is a prefab which generates a mouth shape and animates it in time with audio from a voice clip, in real-time. This gives the illusion of a low polygon character speaking. In addition the NWT script has a number of extra features to help developers bring life to low polygon characters, more easily.

Typically, NWT can be used to create an animating mouth, animating to a character's voice clip, as it plays. This version of NWT works best with low polygon characters with no mouths, we add that part in and animate it for you. You supply the voice clips, or perhaps grab one of our professionally recorded and processed voice packs!

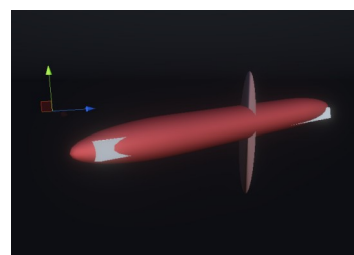
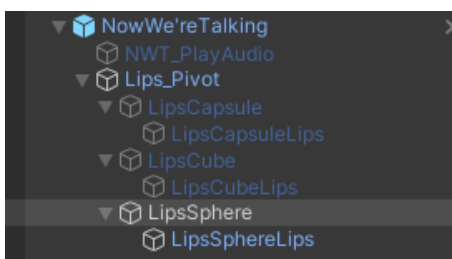
From characters speaking to the player, or to each other, singing, arguing, delivering random or structured phrases and even conversations, NWT is designed to inject more life into your games, cut scenes, stories or films. We hope you'll have a lot of enjoyment using NWT to bring even more life to your own projects by giving your low polygon characters a voice! Well, to be accurate, a mouth (so to speak!).

To speed you on your way, we have also included a small selection of professionally recorded voice clips, from our NWT voice packs, to use in your exploration of the asset as you learn how to use the system. These packs also come with optimised data settings, so the voice clips will work well with NWT mouths, out of the box! Of course you can actually **use any voice clip** which the Unity audio source component can play, but not everyone can record their own voice clips and process them to a high quality.

### NWT Set up

So you've unpacked the NWT Unity Package into your project. NWT is quite simple to set up and initialise, especially when using mouth data presets. We provide a preconfigured prefab, named 'NowWe'reTalking', see image below. To understand the system and how it works, let's take a detailed look at the **Now We're Talking** prefab.

Drag the 'Now We're Talking' prefab from the folder **Assets/NowWe'reTalking/Prefab** in your project and drop it into an empty scene. Select it and open it up to show the prefab's hierarchy.



In the hierarchy it should look like the above left screen grab, in the scene window it should look something like the above right screen grab. The bread stick shape is where each of the current three mouths and 'lips', provided, will appear and animate.

**NowWe'reTalking** – is the parent/root gameobject of the prefab. This parent gameobject holds three components of interest:

- (1) the **NowWe'reTalking** script.
- (2) an animator.
- (3) an audio source.

Let's find out some detail about each of those components. There has been a major update to the functionality here so please read carefully to understand how the changes affect using the prefab.

### (1)\* The **NowWe'reTalking** script

This script is the heart of the system. It allows developers to set parameters which affect the shape, speed, animation and resting pose of the chosen mouth.

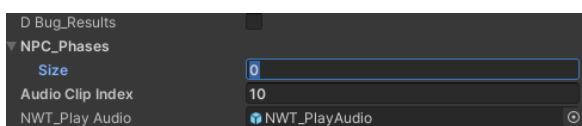
Version **1.03** is a **major update** which allows your characters to store and handle far more complicated conversations than in previous versions of **Now We're Talking!** Each NPC can now have multiple conversations with the player (or other NPC's/none player characters). We call this **NPC\_Phases**. Each phase is a conversation, which means, each phase needs its own list of voice clips which the developer can switch between 'on the fly' using two lines of code (see below).

Having multiple lists of voice clips gives you the opportunity to have NPC's with apparent intelligence and makes it easy to switch between different phases of the game where different voice clip lists can be used. Lets say a game has an NPC who can set a quest for the player. You decide that this quest has four phases, which involves four different conversations between the NPC and the player, as follows:

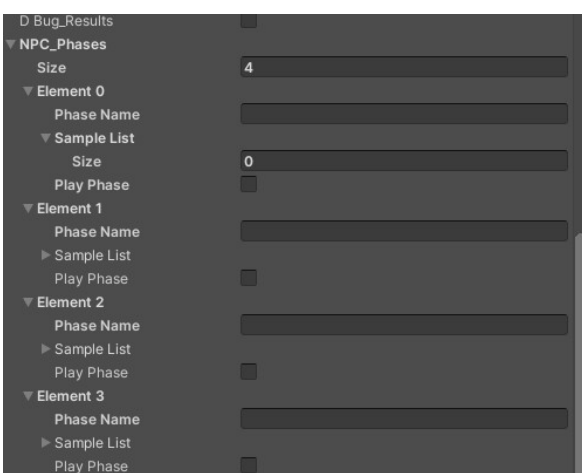
1. a conversation which explains what the quest is that the player must attempt
2. a conversation where the player returns and has failed the quest
3. a conversation where the player returns and has completed the quest
4. a conversation where the player has not yet attempted the quest.

The first **NPC\_Phase** conversation allows the NPC to set the quest for the player. The NPC explains what needs doing to complete the quest during this conversation. This will need a list of voice clips to deliver the quest details to the player and send them on their way. How does it work?

First we need to locate **NPC\_Phases** in the **Now We're Talking!** script on the **NWT\_Prefab** and assign the total number of **NPC\_Phases** for this NPC (you can edit this later). Below is how the Unity Inspector looks before there are any **NPC\_Phases** for this character.



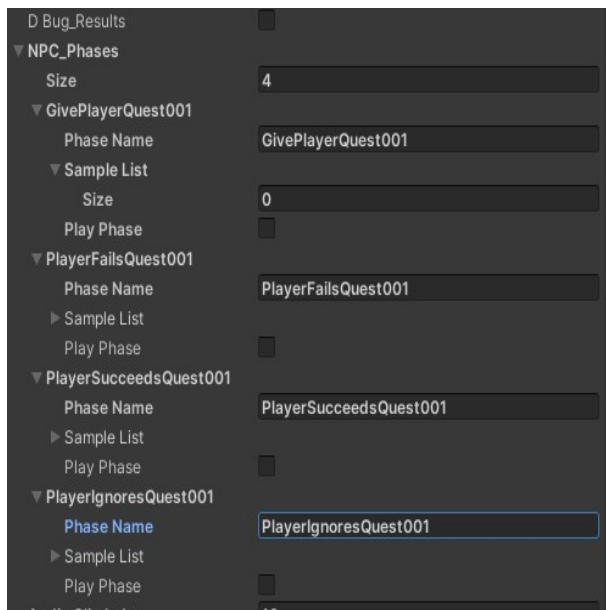
We need four **NPC\_Phases** for this NPC. Set the **Size** to 4, to reveal the four **NPC\_Phases**.



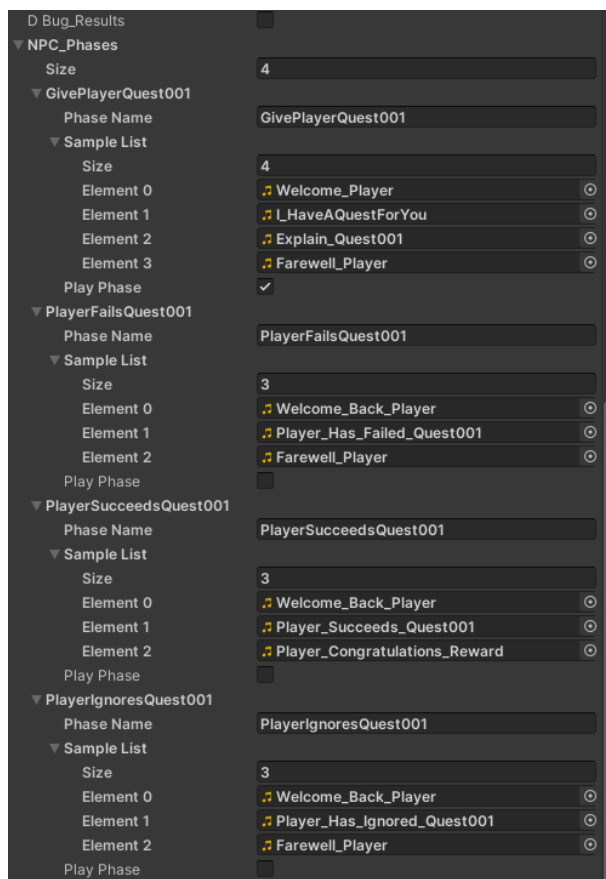
This will expand the **NPC\_Phases** list and set up four empty ‘Sample Lists’ for voice clips. You now have four **NPC\_Phases**, each phase has it’s own Element / Phase Name and a Sample List to go along with it. You now need to give each phase a name you can refer to later, in code. We suggest the name should directly refer to the actual NPC\_Phase:

1. GivePlayerQuest001
2. PlayerFailsQuest001
3. PlayerSucceedsQuest001
4. PlayerIgnoresQuest001

The developer physically types in the empty ‘Phase Name’ text box of each voice clip list in the Unity Inspector, for example ‘**GivePlayerQuest001**’ etc. see below.



Developers now populate each ‘Sample List’ with the correct voice clips for each phase, in order.



It is recommended to carefully name each voice clip so it refers to the content of the clip. This will make it easy to populate voice clip lists and keep everything organised, see above clip names.

Please note that every **NPC\_Phase** has a '**Play Phase**' checkbox underneath the last element in the Sample List. Only one '**Play Phase**' boolean checkbox should be set to TRUE for **Now We're Talking!** to operate correctly. In the above example, the first NPC\_Phase '**GivePlayerQuest001**' is set ready. All the others are unchecked, i.e. false.

If there is only one **NPC\_Phase** on a character then the '**Play Phase**' boolean checkbox can be checked under the first Sample List, in the Unity Inspector and activating that character's '**NWT\_PlayAudio**' gameobject will work as normal, playing each voice clip in order until the **NPC\_Phase** is completed.

If there are more **NPC\_Phases**, as in our example above, where there are four **NPC\_Phases**, developers will need to set the correct '**Play Phase**' boolean for the phase required at any one time.

This is done through code. A second script has been supplied to show how to change the **NPC\_Phase** during a game or story, open the **Now We're Talking/Scripts** folder and look for a script called "setCorrectPhaseFromHere".

The developer uses two lines of code, as shown below. The first line of code initialises a gameobject reference for the NPC with the **Now We're Talking!** Script we want to work with.

```
GameObject nowWereTalkingPrefabOnCharacterToSpeak; //drag the reference to the 'Now Were Talking' character where this script is going to send messages to, to set the correct phase.
```

In the main script, use this call when you wish to set a new **NPC\_Phase**;

```
nowWereTalkingPrefabOnCharacterToSpeak.SendMessage("setTheCorrectPhase",  
"GivePlayerQuest001");
```

The developer uses the second line of code to call the gameobject reference and set the correct **NPC\_Phase**, when they want the phases to change. Use the actual phase name, from the Unity Inspector (we recommend you cut and paste this to avoid spelling errors) use the '**SendMessage**' call replacing the example "GivePlayerQuest001" name with whatever the **NPC\_Phase** name the developer has chosen and wishes to change to.

The main **Now We're Talking!** Script will handle the phase swapping automatically, based on the **NPC\_Phase** name sent from this code. It will also set the other **Phase\_Play** checkboxes to false.

The NWT script will use the **NPC\_Phase** voice clips to animate the mouth based on the slider parameters displayed in the inspector, live, in real-time.

(2) The **animator** should be left in place with its default settings.

(3) The **audio source**

The audio source should be left in its default state and mainly ignored. The audio source on this gameobject will be used by the system to playback audio clips which are fed to it via NWT code.

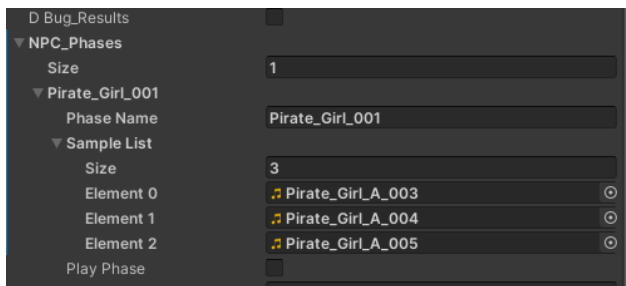
## Quick Start

The NWT prefab has a number of child objects which work together to dynamically change the shape of the three primitives we use, to mimic low polygon character mouth shapes and movements.

The script has a number of sliders which allow developers to make adjustments ‘on the fly’, affecting the current mouth shape during playback. Set the size of the **NPC\_Phases** to 1. Give this ‘Phase’ Element a name such as “Pirate\_Girl\_001”. Populate the Sample List with at least one voice clip (“Pirate\_Girl\_A\_003”) by dragging each audio clip onto the words ‘Sample List’, and releasing.

Alternatively set the **NPC\_Phase** ‘Sample List size’ to 3 and use the filename finder (the file dot) to locate voice clips, and load them into the list as element 0, 1, 2 etc. Add as many voice clips as you need for this NPC conversation. The example shows three voice clips.

If you don’t have any voice clips to hand, we have included a few samples in the audio folder in the Now We’re Talking folder. Under the last element in the Sample List is a checkbox labelled ‘**Play Phase**’. By checking this checkbox (add a tick) you set this **NPC\_Phase** to ‘True’ and when called upon, NWT will use this list as the source of the audio clips to be played, in list order (unless ‘random selection from list yes’ is also checked).



Add a 3D sphere to your scene to represent a character’s head and parent the NWT prefab to the sphere, ensuring the NowWe’reTalking prefab transform position and rotation is set to 0,0,0 and the scale is 1,1,1. In the NowWe’reTalking prefab, select the Lips\_Pivot and move it so it shows on the outside surface of the sphere where you would expect to see a mouth (if it were a head). If you need to, scale up the Lips\_Pivot object to a suitable size. It should now look something like a bread stick on the surface of the sphere. Look at the NWT script component. Set the **Allow Mouth Data Presets** box to true, by checking it (add a tick) and locate the **Mouth Data Presets** slider.

Make sure your main camera is looking at the bread stick like ‘mouth’, from a comfy distance away. Run the Unity Editor. Now find and activate the **NWT\_PlayAudio** game object in the NWT prefab hierarchy in your scene.

In the current configuration, the first voice clip in the list will play. A mouth will be generated and animated in sync with the voice. Once the voice clip has finished playing, the mouth will come to rest in the preset pose (adjustable). To see the next audio voice clip in action deactivate the **NWT\_PlayAudio** game object and reactivate it. Do this until all the clips in the list have been played. The mouth should be responding to each of the speech clips, individually.

When the Unity Editor is run and with the **Allow Mouth Data Presets** unchecked (no tick), the parameter sliders become active and you will be able to make adjustments, knowing you have presets you can always go back to.



To ensure enough time to make adjustments, choose a long voice clip or go to the audio source on the parent gameobject and check (tick) the **loop** check box. Any voice which plays will now loop continuously. With the **Allow Mouth Data Presets** unchecked (no tick) you can make slider adjustments. Please ensure you turn the loop off again later!

When you are happy with your adjustments and **before** you stop the Unity Editor from running, use ‘copy component’ to copy the **Now We’re Talking** component values.

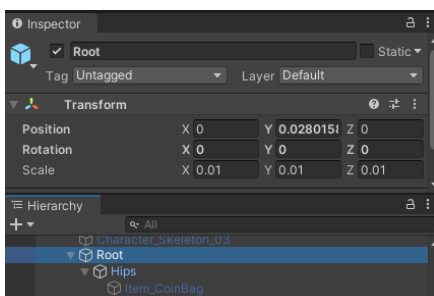
Stop the Unity editor and paste those values back into the **Now We’re Talking** component. Make sure the **Allow Mouth Data Presets** check box is empty so it doesn’t overwrite your new settings. Then save your work. Do this repeatedly to ensure you do not lose settings as you progress.

## Details

To appear on an animated character, the NWT prefab should normally be made a child object of the ‘head’ bone (or the last bone at the top end of the spine) of the low polygon character who will be speaking. Remember, the NWT prefab transform should be positioned and rotated at 0,0,0. Set the **Lips\_Pivot** (the bread stick) as described above, in the approximate position where a mouth would be found on the character. You can position, rotate and scale the **Lips\_Pivot** object to get the best looking result.

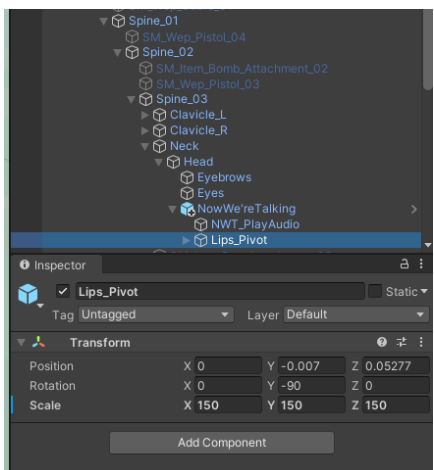
In all cases it’s worth checking the scale setting at the root of the skeleton of your character. If the scale is lower than 1.0, for example, 0.01 then you may not be able to see the **Lips\_Pivot** or bread stick object (or the mouth shapes when running) because changes in scale on a parent object in Unity, also affects child objects.

Look at the image below, the scale is set very low which means all child objects will reflect this scale, in some cases the child object maybe too small to see.



In this case you may have to increase the scale of the **Lips\_Pivot** to 100 - 150 on each axis, to compensate. Try different values and choose one which best fits the low polygon characters you are using.

The NWT script on the prefab holds the references to the three primitive objects used to create animated mouths for the system, **LipsSphere**, **LipsCapsule**, **LipsCube**.



These are children of the **Lips\_Pivot** gameobject. Use the **Lips\_Pivot** object to manipulate the starting position and rotation of the mouth objects to your preference.

The three mouth shapes need to appear slightly in front of the character’s head in the position where a mouth would be. It is important to use the **Lips\_Pivot** for major positioning and rotation.

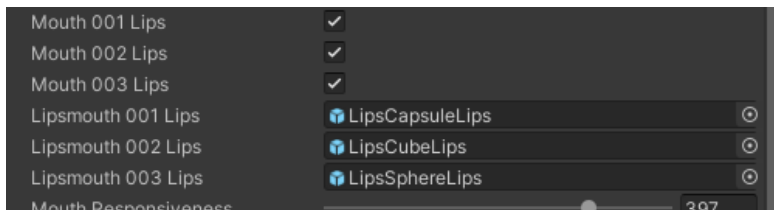
Small adjustments can then be made to the three mouth objects, **LipsSphere**, **LipsCapsule**, **LipsCube**, but only fine tuning of position and rotation – do not adjust scale on these three mouth shapes.

In the above example, the root object scale was 0.01 so we compensated to 150 on the **Lips\_Pivot**.

In addition there are slots for game objects called **MyEyes** and **MyEyeBrows**. If your low polygon character has these as separate meshes (see above hierarchy image), you can drag and drop each reference into the appropriate slot. (Note there's an 'on/off' check box further down the NWT script for each called **Emotive Blink On**, **Just Blink On** and **Eye Brows On**. Don't forget to set these on if you want to have some animation on them, see Blinks checkboxes diagram below).

Next is a reference to the audio source on this gameobject. Please leave this as set. It should automatically be set to the audio source on the parent **NowWe'reTalking** game object.

This component also has reference slots for the supplied mouths to each have Lips. These can be checked on or off here, they are off by default.



Lips can have materials set directly in the hierarchy too, click on the **LipsCapsuleLips** object in the inspector to reveal it in the hierarchy and click on it directly to navigate to the material slot in the inspector. Change the material to your preferred choice or duplicate a material and add your own colour.

The next six sliders set the various parameters for tuning the animation of your chosen mouth shape to the voice clip. These will be examined in more detail, far below.

Next are four more sliders. **Mouth Resting Threshold**, **X Pose Scale**, **Y Pose Scale**, **Z Pose Scale**, let's discuss these in more detail.

### **Mouth Resting Threshold**

Voice clips vary in volume. At a low volume or during silence you may wish your character's mouth to go to a resting pose, closed but still visible. The threshold slider allows you fine control over what volume level the mouth will rest. This slider is useful for preventing jittery mouths, forcing mouth shapes to adopt the resting shape which is set using the following three settings. **X Pose Scale**, **Y Pose Scale**, **Z Pose Scale**.

### **X Pose Scale, Y Pose Scale, Z Pose Scale**

These three settings allow you to set the resting pose of the mouth shape chosen, when the mouth stops moving. During silences or gaps between words the mouth will automatically rest based on the **Mouth Resting Threshold**. Mouths dynamically change shape and size during speaking, NWT tries to mimic these dynamic changes for a more realistic result.

### **Mouth Shape**

This slider lets you dynamically switch the mouth shapes to one of three basic shapes. **LipsSphere**, **LipsCapsule**, **LipsCube**.

Although these are Unity primitive shapes, they make excellent, low polygon mouth shapes for low polygon character's mouths. With careful selection, parameter adjustment and colouration it is possible to have a whole crowd of different looking characters all speaking, see out 'Now you can see the people sing' video on the Chatterware YouTube channel. Another example, using the 'teeth'

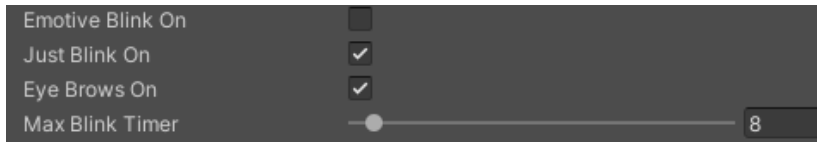


material with the **LipsCube** on a bearded character can be very effective, when finely tuned – see the two different ‘Bridge of Destiny’ clips on YouTube for an example of this.

Try the different mouth shapes and varied material colours on each character to see which suits best.

**Lip Reducer** is a setting to assist with reducing the overall movement of the chosen lips, it can be useful to reduce movement spikes and dramatic shape changes.

Blinks checkboxes are shown below.



**Emotive Blink On** sets the character’s eyes into a more emotive mode, moving slowly between poses. **Emotive Blink On** has PRIORITY over **Blink On** and will always override it, so switching **Emotive Blink On** during a more emotive speech will stop your character from actual blinking and instead their eyes will have more ‘emotion’ in their movement.

**Just Blink On** sets your character’s eyes to a very simple fully open or almost closed, blink. This setting is overridden when Emotive Blink On is set to ‘True’ (ticked).

**Eye Brows On** sets the character’s eye brows moving in a randomly expressive motion.

**Max Blink Timer** slider sets the maximum time up to which a random blink will occur, (0 to ‘maximum time’).

## Complex

NWT is simple in principle however it can be quite complex to fine tune parameters from scratch successfully, until you understand how to use the sliders to maximum effect, for each mouth shape and voice clip.

For this reason we have built in some optimum **Mouth Data Presets** which you can use with your voice clips and your chosen mouth shapes, to get a character’s voice and mouth working very quickly, out of the box.

Using the **Mouth Shape** slider on its own, keeps all the parameter slider data you have already set and simply changes the mouth shape used to generate the visual mouth using a different primitive shape. This is non destructive as far as settings go.

If you set the **Allow Mouth Data Presets** check box (so it is ticked) you will link the data presets slider, just below, to the mouth shape slider. Warning, this is a destructive mode. All your settings will instantly be lost, swapping to the NWT standard presets, for each individual mouth shape.

BEWARE if you haven’t made a note of **your** current settings or if you have not used ‘component copy’ on the ‘Now We’re Talking’ component, **you will lose all your settings**. Please be careful not to lose your hard work.

Linking the **Mouth Data Presets** slider to the **Mouth Shape** slider also overrides the **Mouth Shape** slider control which you will no longer be able to move directly. Now use the **Mouth Data Presets**

slider and this will automatically move the **Mouth Shape** slider as well, between the three sets of preset data.

If you then wish to make additional changes to the settings, you can then un-check the **Allow Mouth Data Presets** and begin making your own adjustments. Once you are happy make sure to copy the NWT component data. Stop the editor. Then paste the component values back into the NWT component and ensure the **Allow Mouth Data Presets** check box is empty – or it will destructively reload presets on running the editor!

### How it works

**Now We're Talking** works by playing voice clips from a preset list which the developer loads in to the NWT component, in the Unity inspector, under **NPC\_Phases**.

To play each voice clip in order from the current phase list, simply activate the **NWT\_PlayAudio** gameobject on that character (it's part of the NWT prefab). NWT will do the rest. Voice clips normally play in top to bottom order. For a randomised order of playback, ensure '**Random Selection From List Yes**' is checked (ticked).

In addition there's a '**Welcome**' voice clip which can be assigned and played once, the first time this character is met. This is controlled by a boolean variable named '**welcomeYes**'. See below for details.

Locate the **NPC\_Phases** list in the **Now We're Talking** script component, notice a small triangle pointing to the right.

Click on the triangle to reveal the Size is 0 there are no phases and no clips in the phase list (to begin, the list is empty). To work with NWT this list must contain at least one **NPC\_Phase** and that should have at least one voice clip in the Sample List. Name the **NPC\_Phase** however you wish, and drag and drop any number of voice clips required, into the words Sample List, in the order they should play. The developer should load this list with all of the voice clips required to be spoken by this character, in this **NPC\_Phase**. Under the last element of the Sample List is a 'Play Phase' checkbox. Ensure this is checked (add a tick) to allow NWT to operate correctly.

If you have an npc who has four random lines he or she can deliver, load those four voice clips into the list in any order. Set the **Random Selection From List Yes** check box (ensure it is ticked). When activated, this will now randomise the playback order of the clips and play the selected, random voice clip. Leave this check box unticked if you want to play the voice clips in the order they appear in the list, top to bottom.

To make the character speak the next voice clip in the list, simply activate the **NWT\_PlayAudio** gameobject, locate it under the parent **Now We're Talking** game object, either in the timeline or via code (see suggested trigger volume example below).

By default this gameobject is deactivated, when you want the character connected to it to speak, activate it, then deactivate it again to prepare for the next activation and next line of speech.

In code terms the **NWT\_PlayAudio** gameobject could be activated or deactivated when a player enters a trigger volume or exits a trigger volume, as follows.

```
NWT_PlayAudio.SetActive(true);  
and  
NWT_PlayAudio.SetActive(false);
```

The gameobject can also be activated using the Timeline.

### Using Timeline

Using timeline, a developer can quickly create a cutscene conversation between any number of characters. Each character has their own **NPC\_Phase** list (as described in detail above) containing their speech clips, in order. Under the last element of the character's Sample List is a 'Play Phase' checkbox. Ensure this is checked (add a tick) to allow NWT to operate correctly.

Simply drag the **NWT\_PlayAudio** gameobject from the character who is next to speak, onto the timeline as an 'Activation Track' at the appropriate position on the timeline and the character will speak when the timeline playback head reaches that activation clip. NWT expects one activation clip for each voice clip required. This will activate the voice clips in order of the list so the conversation plays, in steps, unless you have set the list to be randomised. With multiple characters, it is fully possible to set up alternate activations for each character, to mimic a conversation.

Timing is essential and you should allow enough time between each activation so that each speech clip creates the illusion of a well paced conversation. The length of speech clips is shown in the inspector in the editor. This could also be used to have characters sing a song together or they could have a huge argument or a gentle philosophical discussion about a quest, or anything you need for your game. The use is almost infinite.

### NPC\_Phases

This is the list of lists, holding all of the voice clips in separate **NPC\_Phases** for a character. You can load them either in order (for a conversation) or no specific order, for random speech.

There are a couple of extra special settings for these clips.

### Random Selection From List Yes

Setting this check box (so it is ticked) will randomise the playback from the list of voice clips your character has in their **NPC\_Phases** clip list. This means an npc can be setup with lots of different phrases, the system would randomise playback from the full list of voice clips.

### Random Selection From List

This number is an integer and shows the number of the clip chosen for play back at this time. It's useful for debugging activations.

**Welcome Yes** check box. When checked (ticked), the first phrase this character will speak is the **Welcome Clip**, which the developer needs to drag and drop into the special **Welcome Clip** slot in the inspector. This **Welcome Clip** will only play once, the first time the character speaks and it is not part of the randomised clip list. That list will play after the welcome clip is finished, then continue in clip order or in random order depending upon character settings. This **Welcome Clip** could be a special greeting or welcome, or phrase you only want to use once.

### NWT\_PlayAudio

This is the key gameobject used to activate the playback of voice clips lists in the **NPC\_Phases** list. It can be used from code, perhaps using an IEnumerator, a trigger collider or on the timeline to play a series of voice clips which would be activated using an activation track. Effectively, every time this gameobject is activated it plays a voice clip from the preset list.

There are a number of tutorials on our website showing examples of each of the three mouth shapes we support currently.

## **The Six Sliders of Doom!**

If you decide to start from scratch, you need to concentrate on the top six sliders, which are as follows:

**Mouth Responsiveness, Lip Speed, Lip Speed Fine, Size Factor, Min Size, Max Size.**

Initially slide all sliders to the left and gradually move them to the right, start at the top and work down. Different mouth shapes react differently to these sliders, so the only way to figure out how to set them correctly and how that even small changes can have huge effects on balancing them, is to practise. Balancing the settings for different mouths is how to get the best from them. You will soon discover the complex way these settings are devised to allow the developer to tune and control the mouth shapes for a character's voice clips. It's also important to note that voice clips should be consistent for characters.

Also look at the Mouth Data Presets we provide to give some idea as to where the sliders work best in each mouth shape case. Use these as a starting point.

In some cases the mouth shapes seem to distort strangely, this is normal as each shape reacts slightly differently to the combination of setting in these sliders. At this point in your set up, move the sliders slowly and carefully as you tune the balance, where the shapes change.

Work with **Size Factor, Min Size and Max Size** as a group of settings, and work with **Mouth Responsiveness, Lip Speed and Lip Speed Fine**, initially in that order as a group.

## **Which Mouth Shape to use?**

Not all low polygon characters will be able to use our system due to the fact we overlay character's faces. Those with meshes already sculpted into a mouth shape are unlikely to be successful with our system. We are continuing to develop **Now We're Talking!** to try to accommodate as many low polygon characters available.

The reason we have provided three different mouth shapes to use with this first release of **Now We're Talking!** is because some variation amongst characters is good and not all **mouth shapes** will work on all characters. We hope to add new shapes to our asset as time progresses to allow for a greater range of characters to be useable with our asset.

## **Extra Bonus Feature – Sci-Fi comes to Now We're Talking! - now robots can talk too!**

We've just added a fun **Sci-Fi** bonus feature to the system. Now we can have robots talking too! With this new feature comes the ability to select two emissive materials to swap between, to give the impression of robots speaking. By flashing lights on / lights off. Think about how you know Daleks are speaking! Their lights, light up as they speak, indicating which Dalek is speaking.

At Chatterware, we've gone one better. You can select the game objects to affect. We provide a number of emissive materials to try. Using contrasting colours works well, red and green are a good example. You must set the check box (add a tick) called '**Flash Yes**' to ensure it will work.

When the robot character's voice clip plays the 'I'm speaking' indicator lights can be set to flash on and off. The developer has a sensitivity control too. The combination of a moving mouth which lights up and changes colour as the character is speaking, is compelling. See the Chatterware YouTube channel for a video tutorial on this and the resultant video.

## FAQ

(1) The character always speaks the first phrase when the game starts.

Hint: Make sure the **NWT\_PlayAudio** game object defaults to deactivated on all characters you do not wish to speak. In timeline based conversations ensure the **Audio Clip Index** is set to -1.

(2) If the mouth is jittery try adjusting the **Mouth Resting Threshold** or try changing **mouth shapes** first then try altering the **Mouth Responsiveness**, **Lip Speed** and **Lip Speed Fine** settings. Don't forget to uncheck the **Allow Mouth Data Presets** box so you can make adjustments to the settings.

(3) Why do the mouth shape settings work on some voice clips and not on others? This happens generally when the consistency of the volume and noise is varied between sound clips. We recommend professionally recorded voice clips which are processed to be clean and crisp. This way the settings should work consistently for each voice clip.

(4) Do you have a support process or a contact email? Yes

(5) I've set it all up but it isn't working! Have you checked the 'Play Phase' checkbox (add a tick) under the last voice clip in the Sample List you wish to play? Alternatively have you set that up using code?

(6) Is there an easy way to DEBUG my settings ? Yes. Just above **NPC\_Phases** is a **D Bug Results** checkbox – add a tick in there and run the editor again. Open the console to see reports of what may be going wrong!

Our website is: [www.chatterware.com](http://www.chatterware.com)

YouTube channel Tutorials: <https://chatterware.com/tutorials/>

Support email is: [support@chatterware.com](mailto:support@chatterware.com)

Also please join our Discord channel: <https://discord.gg/U86cu3jcjp>

Our Discord channel is where our community can help each other and where you can request new features and voice pack themes.