QUICKSTART GUIDE

INTRODUCTION

The Virtual Human Project (VHP) provides developers with a scalable, easy to integrate and optimized package to procedurally animate realistic virtual humans in Unity. Following this guide, you will learn how to create a new preset using the dedicated tool to configure, save and edit your characters' facial expressions. Then, you will learn how to configure the main components of the toolkit to add emotions, several gaze strategies and lip synchronization.

PREREQUISITES

This toolkit has been developed for **Unity 2019.4** or higher. To benefit from its full potential, you will need **rigged characters with a proper set of blend shapes**. Two demo characters (female and male) created using Character Creator 3 are provided with the project. The demo project relies on, but is not restricted to, Unity High Definition Render Pipeline 7.4.1.

TABLE OF CONTENTS

CONTACT

Geoffrey Gorisse, PhD
geoffrey.gorisse@gmail.com
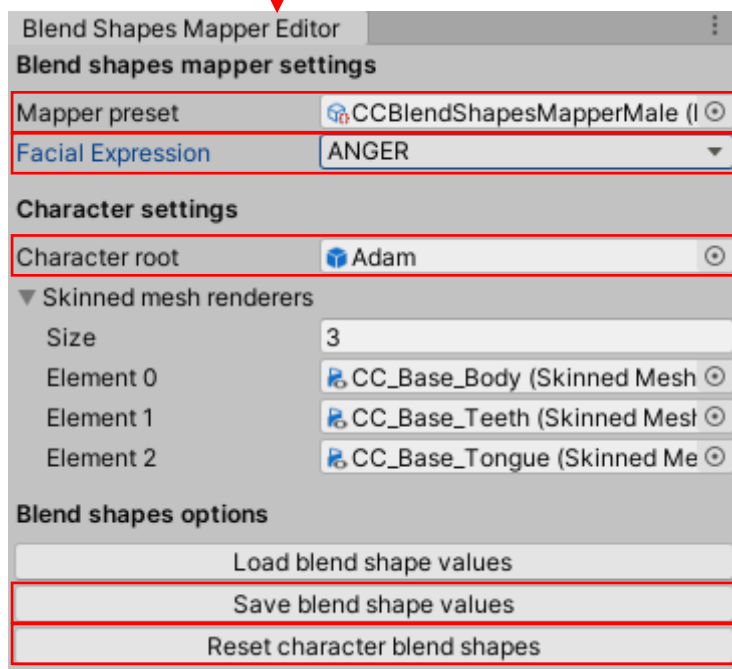
BLEND SHAPES PRESET

The first step, after importing your character, is to create a new preset to configure facial expressions used by the runtime scripts for procedural animations.

    I.   Right click on the project window -> Create -> Virtual Human Project -> Blend Shapes Mapper.

   II.   Click on Window -> Virtual Human Project -> Blend Shapes Mapper Editor.

  III.   Drag and drop the mapper preset created in step 1 (1).

  IV.   Select the facial expression to be edited (2).

   V.   Drag and drop the root gameObject of the character (3). Skinned meshes with blend shapes should appear in a list.

  VI.   Adjust the blend shapes values to match the selected facial expression (4).

 VII.   Save the current values using **Save blend shape values** button (5).

VIII.   After editing and saving the character's blend shapes, you can reset the values to configure a new facial expression from scratch using the **Reset character blend shapes** button (6).

Note: The created preset can be used for every character based on the same set of blend shapes. However, depending on your character's morphology, you may need to adjust its facial expressions. To edit a preset, create a copy of the existing scriptable object and use the **Load blend shape values** button of the blend shape mapper editor tool to edit the current values following the process described above.
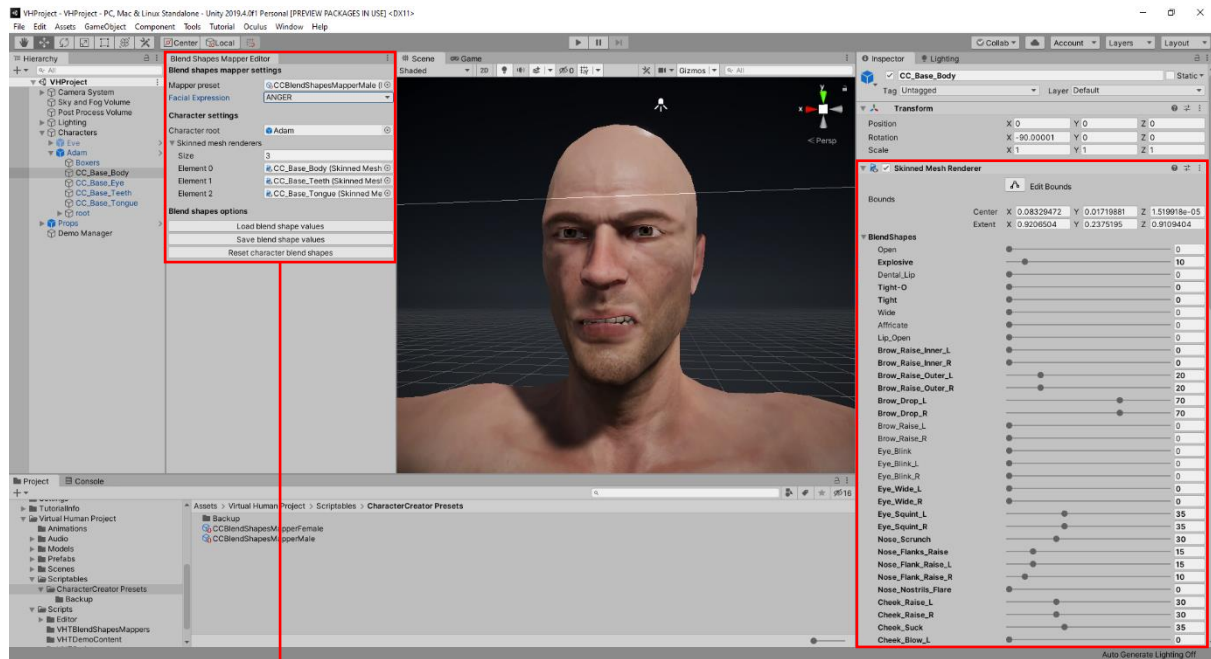
(4) Adjust blend shapes values

Blend Shapes Mapper Editor

Blend shapes mapper settings
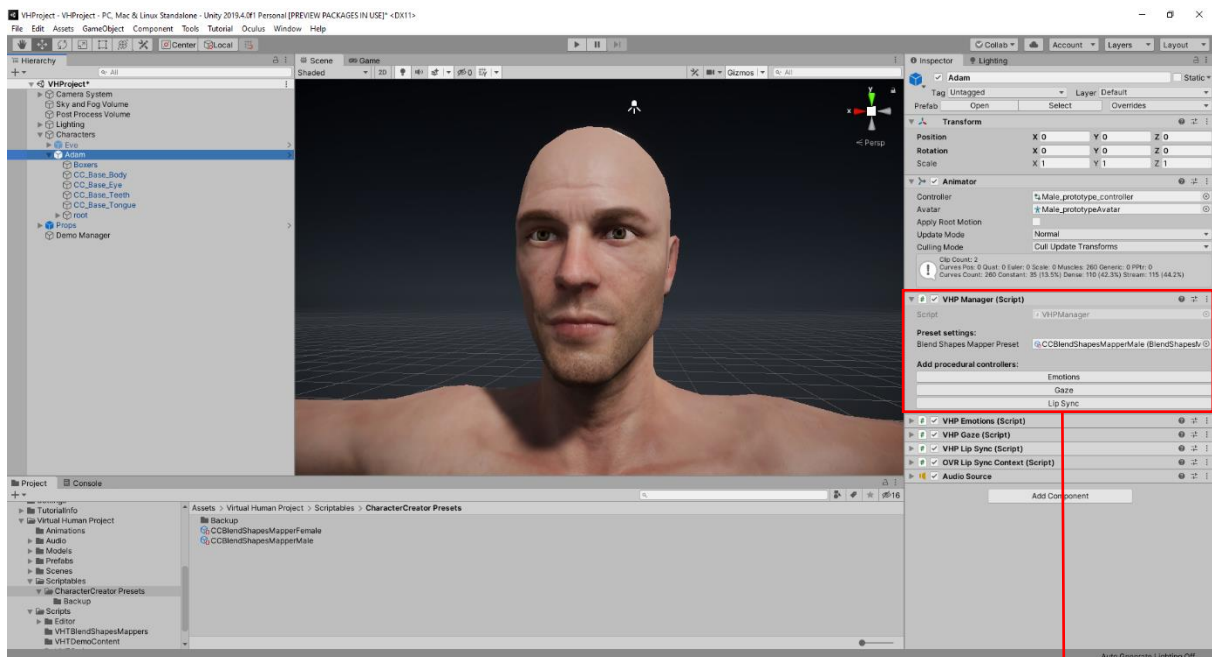
Mapper preset   CCBlendShapesMapperMale (I   →   (1) Blend shapes mapper preset

Facial Expression   ANGER   →   (2) Facial expression to edit

Character settings

Character root   Adam   →   (3) Character root gameObject

▼ Skinned mesh renderers

   Size   3

   Element 0   CC_Base_Body (Skinned Mesh

   Element 1   CC_Base_Teeth (Skinned Mesl

   Element 2   CC_Base_Tongue (Skinned Me

Blend shapes options

Load blend shape values

Save blend shape values   →   (5) Save current blend shapes

Reset character blend shapes   →   (6) Reset blend shapes

4

CHARACTER MANAGER

The first step toward animating your character is to add a **VHPManager** component to its root gameObject. Once the script is added to your character you can drag and drop the character's mapper preset created previously (1). Then, click on the buttons to add the needed **emotions**, **gaze** and **lip sync** controllers (2). Other required components will be automatically instantiated when adding controllers.
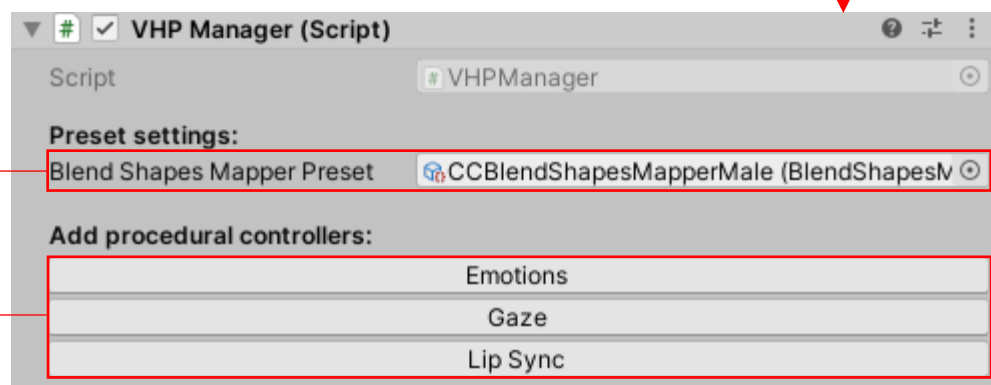
EMOTIONS

The **VHPEmotions** script allows to change your character's facial expression according to Ekman basic emotions[1]. You can change emotions manually in Play mode using the sliders in the inspector window, this should dynamically load the preset values you created following the guidelines provided in the previous section.

The public emotion fields are accessible through code to allow you to change your character's emotional state depending on the content of your application. Keep in mind that emotion values are mutually exclusive and that you will not be able to turn on multiple emotions at the same time.

(1) Emotion sliders

---
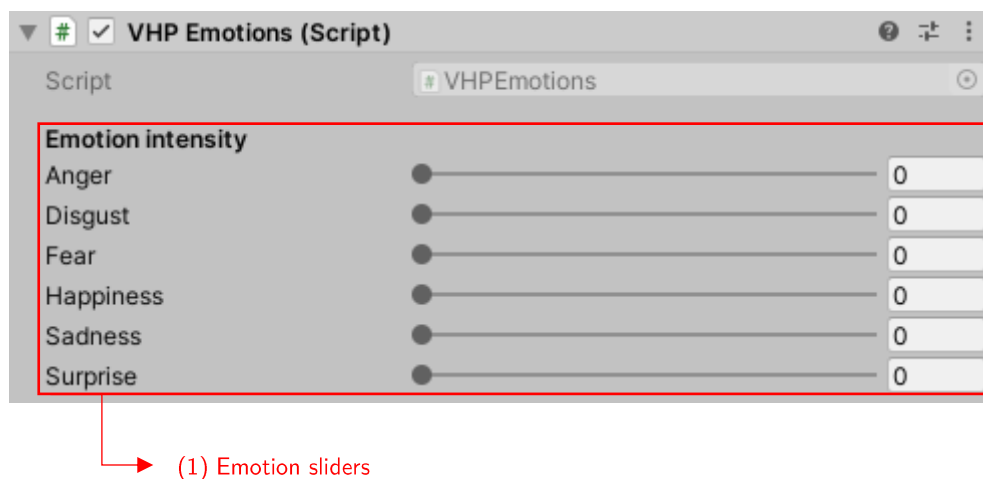
[1] Ekman, P. (1999). Basic emotions. *Handbook of cognition and emotion*, *98*(45-60), 16.

GAZE

Using the **VHPGaze** script, you can apply one of the four gaze modes. To make the toolkit compatible with several rigging templates, it is necessary to add the character's head bone and both left and right eye bones (1) as well as their forward axis direction (2). Then, one of the gaze modes can be selected (3):

1. **Probabilistic**: automatic probabilistic target ponderation based on distance, sound proximity and volume, and movement velocity (scene configuration required and described in the next section).
2. **Random**: gaze orientation modification with variable frequency and duration based on random positions in the field of view of the character.
3. **Static**: static gaze orientation matching the forward direction of the character.
4. **Scripted**: single scriptable target position.

Gaze micro-variations and blinking options can be turned on if needed without affecting the selected gaze mode (4). Finally, turning on the agent mode (5) will enable inverse kinematics to override the character's upper body rotation depending on the target location (IK pass must be enabled in the animator settings).
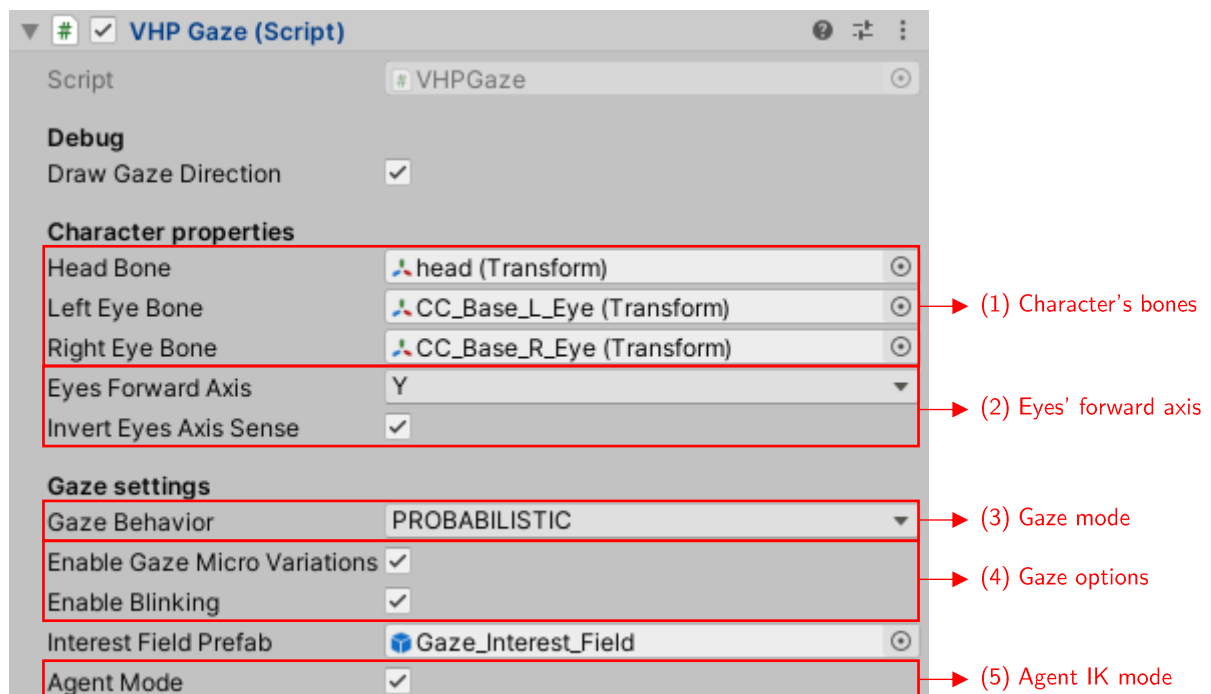
PROBABILISTIC GAZE CONFIGURATION

If you plan to use the probabilistic gaze mode, you must configure the scene first.

I.   Click on Window -> Virtual Human Project -> Probabilistic Gaze Configurator.

II. The Gaze_Target prefab (1) should be automatically added from the project resources.

III. Select all the 3D objects to be considered as potential targets from the hierarchy window (2).

IV. Add the selected gameObjects to the target list using the **Add selected objects** button (3).

V.   Instantiate the target prefab using the **Add target Prefab** button (4). A child gameObject will be added with a collider scaled to the bounds' size of the mesh renderer. You can also add other targets manually if you need an object to have several targets.

Note: The collision matrix of the project will be automatically updated to disable collisions between targets and other gameObjects in the scene. All instantiated targets can be removed from the scene using the **Remove all target** button (5).

(2) Potential 3D targets

Gaze target settings

Gaze target prefab   🟦 Gaze_Target   ⊙   ——————————→ (1) Gaze target prefab

▼ Gaze targets

Size   7

Element 0   ⚒ Static_Target_Circle ⊙

Element 1   ⚒ Static_Target_Circle ⊙

Element 2   ⚒ Dynamic_Target_Circ ⊙

Element 3   ⚒ Static_Target_Circle ⊙

Element 4   ⚒ Sound_Target_Circle ⊙

Element 5   ⚒ Static_Target_Circle ⊙

Element 6   ⚒ Static_Target_Circle ⊙
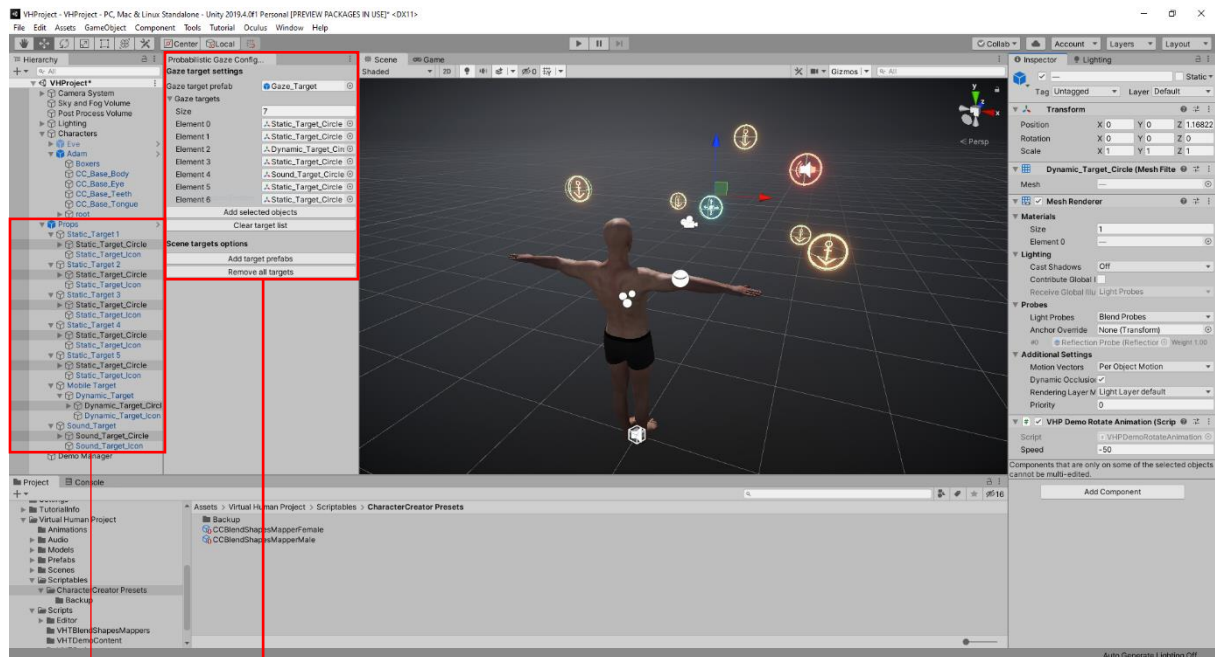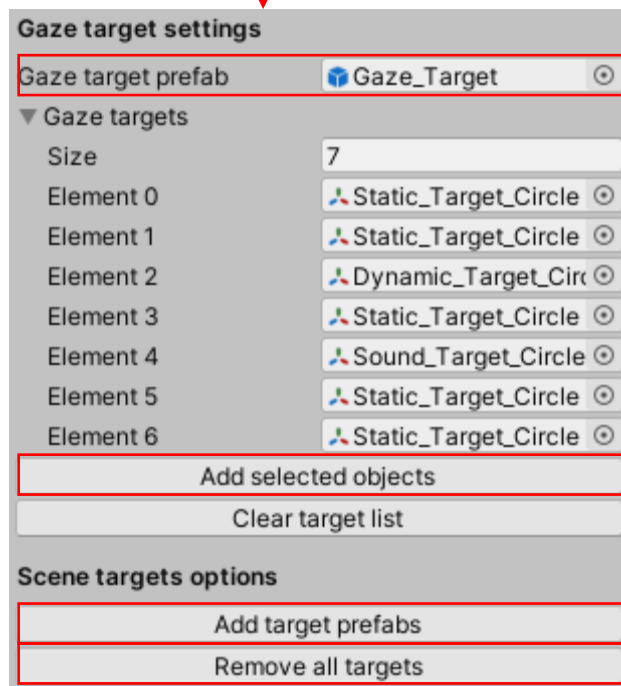
Add selected objects   ——————————→ (3) Add selected objects to the list

Clear target list

Scene targets options

Add target prefabs   ——————————→ (4) Target prefab instantiation

Remove all targets   ——————————→ (5) Remove targets from the scene
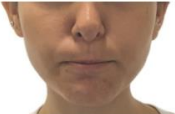
LIP SYNCHRONIZATION

The **VHPLipSync** script allows to use lip synchronization based on both pre-recorded audio clips and real-time microphone input. You can select the lip synchronization mode using the **Lip Sync Mode** public field (1). Do not forget to turn on the OVR **Audio Loopback** (2) if you need to hear your microphone in play mode.

Note: To properly set your viseme presets, please refer to the reference card available in the appendix (also available as a PDF file in the project folder).
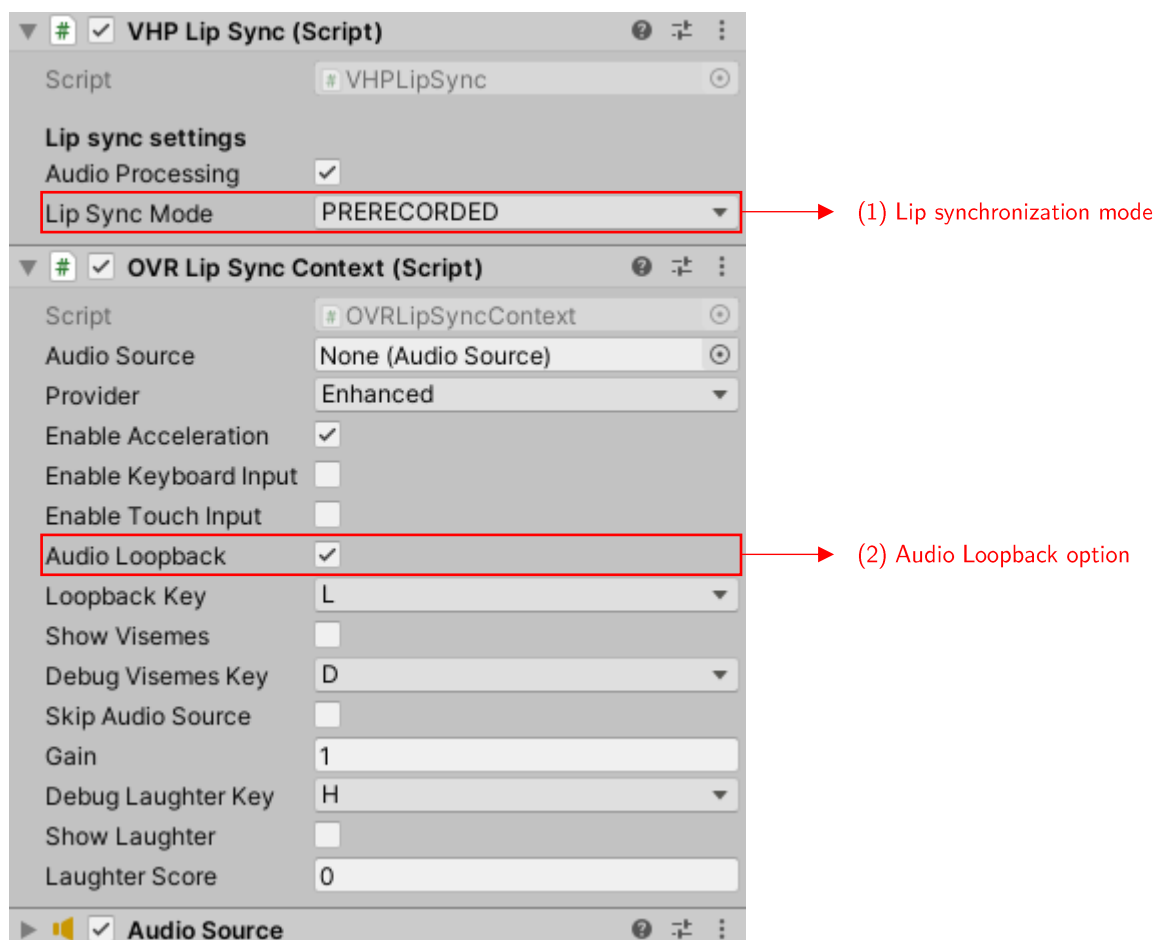
APPENDIX: VISEMES REFERENCE CARD

Viseme Name	Phonemes	Examples	Mild Production	Emphasized Production	3/4 Rotation
sil	neutral	(none - silence)		None	
PP	p, b, m	put, bat, mat			
FF	f, v	fat, vat			
TH	th	think, that			
DD	t, d	tip, doll			
kk	k, g	call, gas			
CH	tS, dZ, S	chair, join, she			
SS	s, z	sir, zeal			
nn	n, l	lot, not			
RR	r	red			
aa	A:	car			

aa	A:	car	
E	e	bed	
I	ih	tip	
O	oh	toe	
U	ou	book	