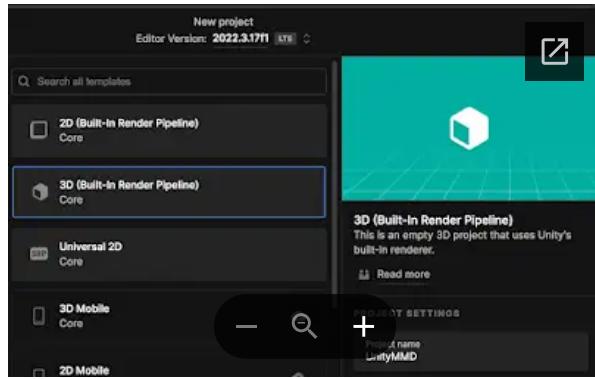


UnityでMMDモデルを使ってMVを作成する時の忘備録 Unity編

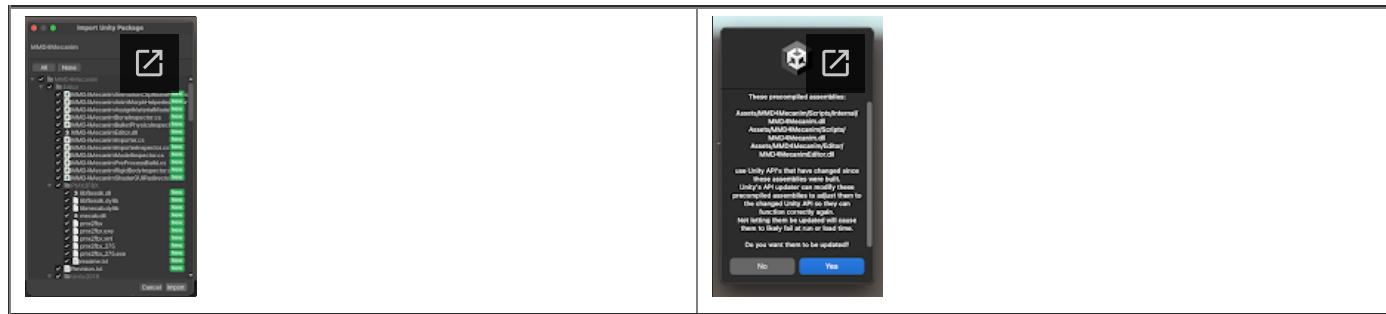
1. Unityと各種ツールの導入

Unityをまずインストールしなきゃいけないのですが、Blenderとは違い登録などが必要なので具体的なインストール方法はさすがに冗長なんでここには書きません。あと、Unityは有料プランなどもあるのですが無料版のUnity Personalで大丈夫です。とりあえず登録を済ませてUnity Hubが立ち上がるようになったらInstall EditorからLTSのバージョンをインストールしてください。ちなみにこれを書いてる時に使用しているエディタのバージョンはUnity*(2023.17f1)LTSでした。

エディタをインストールしたら次にUnity Hubの右上にある**New Project**を選び3D(Built-In Render Pipeline)を選択してください。Project Nameは好きなものでかまいません。

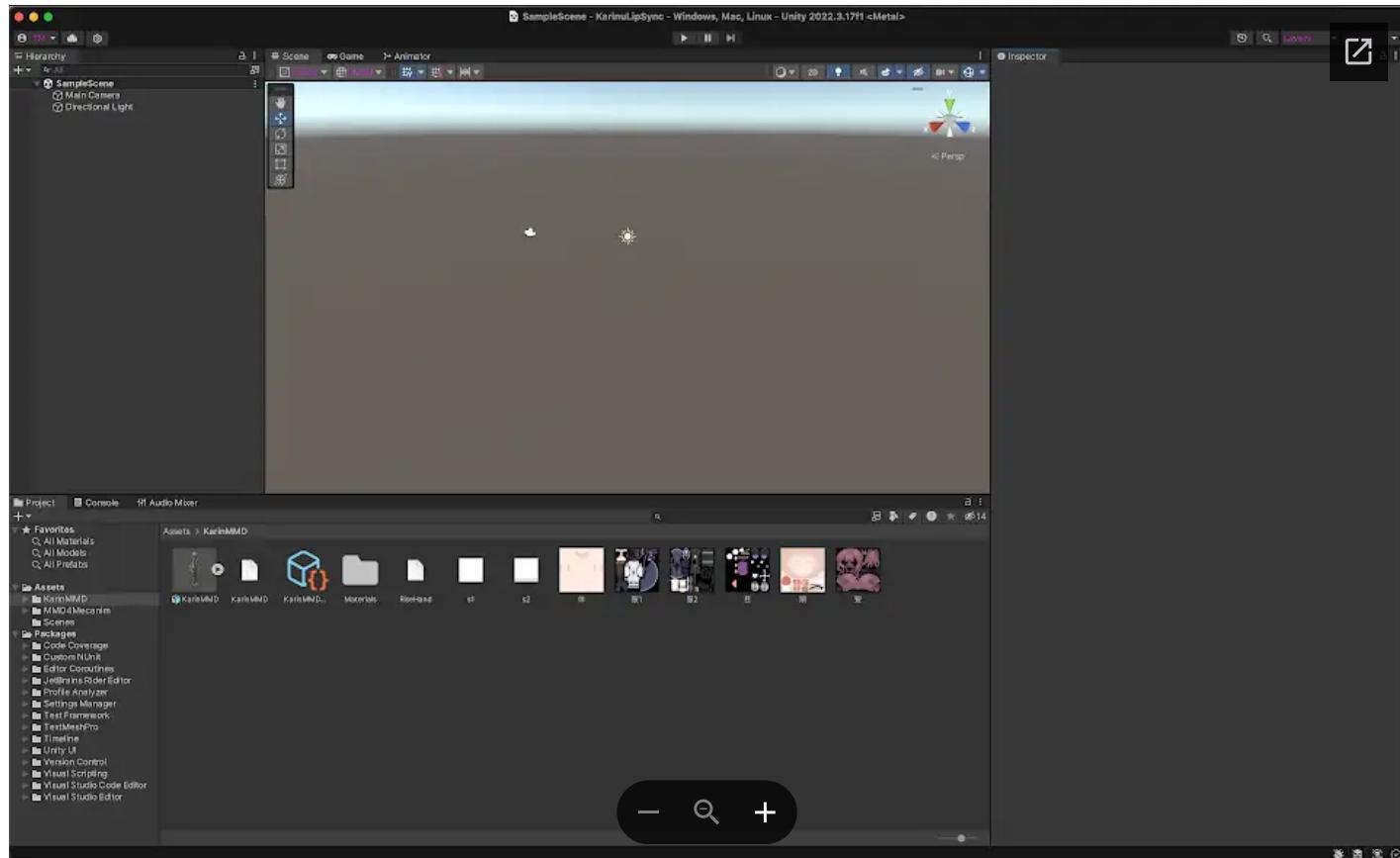


Unity Editorが起動したら次にNoraさんが開発された[MMD4Mecanim \(Beta\)](#)を導入します。これはMMDのデータをUnityの形式に変換してくれる大変便利なツールです。ダウンロードした[MMD4Mecanim_Beta_20201105.zip](#)を解凍したら中に入っているMMD4Mecanim.unitypackageを開くとエディタにこのようなダイアログが表示されると思うので**import**または**Yes**を選択しインストールしてください。



時々インストールされたものがうまく認識されなかったりするのでここでUnityのエディタを再起動しておくといいかもしれません。

次に左下のAssetsの中にBlenderで作成していたKarinMMDフォルダをドラッグ&ドロップで入れておきます。フォルダの中身は下記の画像といっしょになっていれば間違ってない思います。もし足りないファイルがあるとしたらBlenderのエクスポートで手順の間違いがあると思うのでもう一度確かめて見てください。

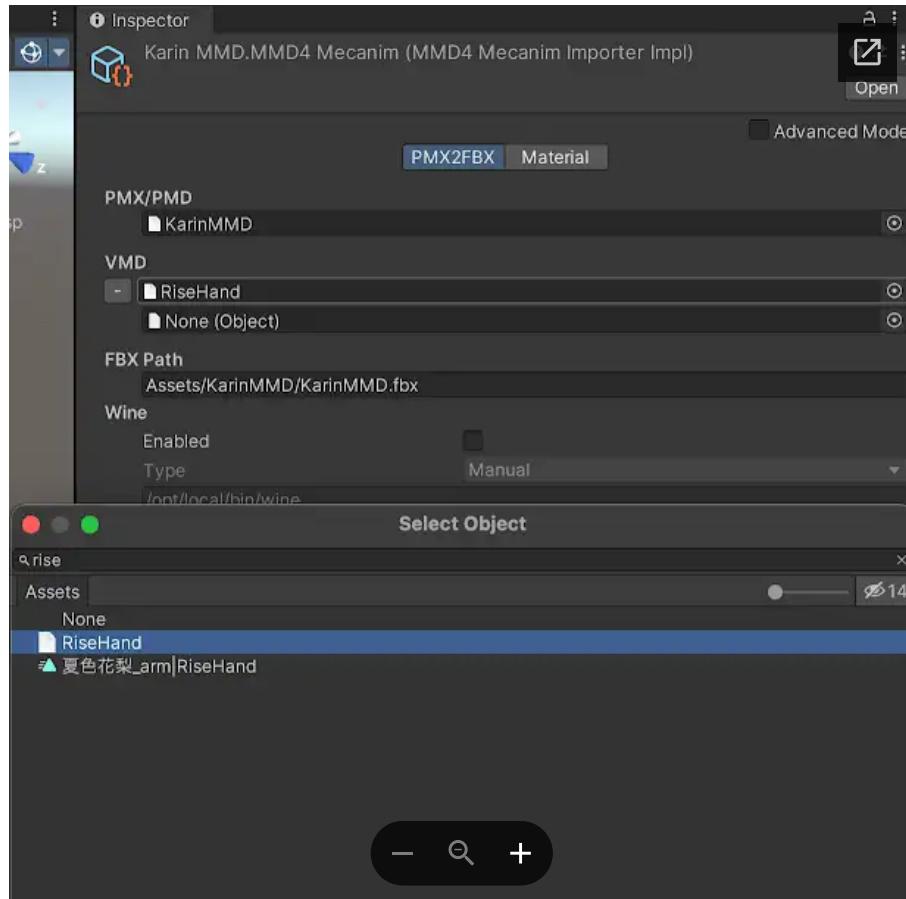


次に先ほどドラッグ&ドロップで入れたKarinMMDフォルダ内のKarinMMD.pmxを選択すると画面右のインスペクタに規約の同意を求められるのでチェックし同意します。



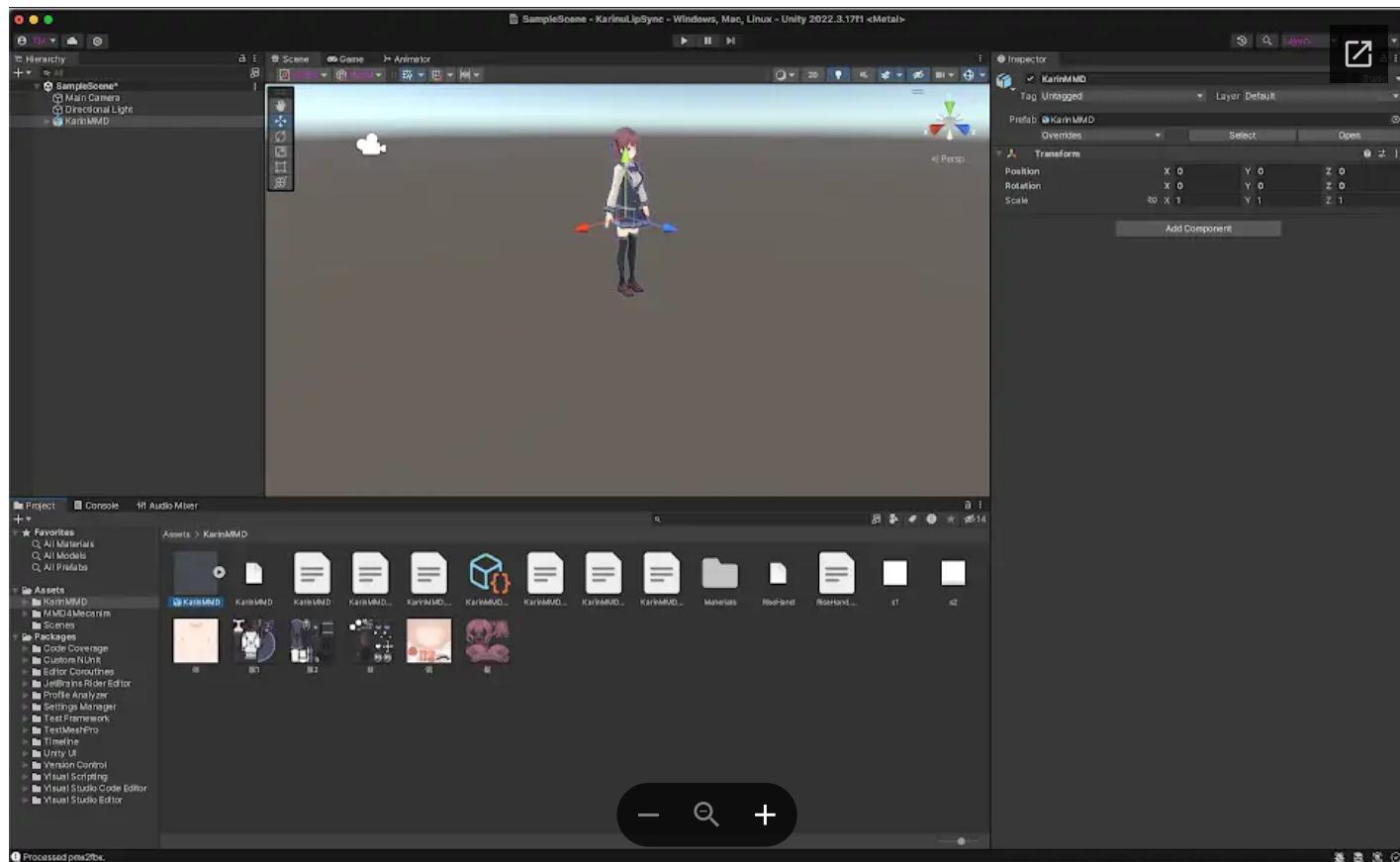
同意したら次の画面のようにVMD欄のNone(Object)となっている右側の白い円をクリックし、虫眼鏡マークのサーチ欄にriseと入力し絞られた結果からRiseHand.vmd(KarinMMDフォルダ内にあるBlenderでエクスポートしたファイル)を指定します。

ここでVMDファイルを指定することでMMDのpmxファイルをUnityが扱えるfbxファイルに変換する時にMMD用のアニメーションデータも一緒に変換されるので、新しいアニメーションを追加する場合はこのVMDファイルをその都度指定する作業が必要です。指定したら右下のProcessを押してください。変換が始めます。



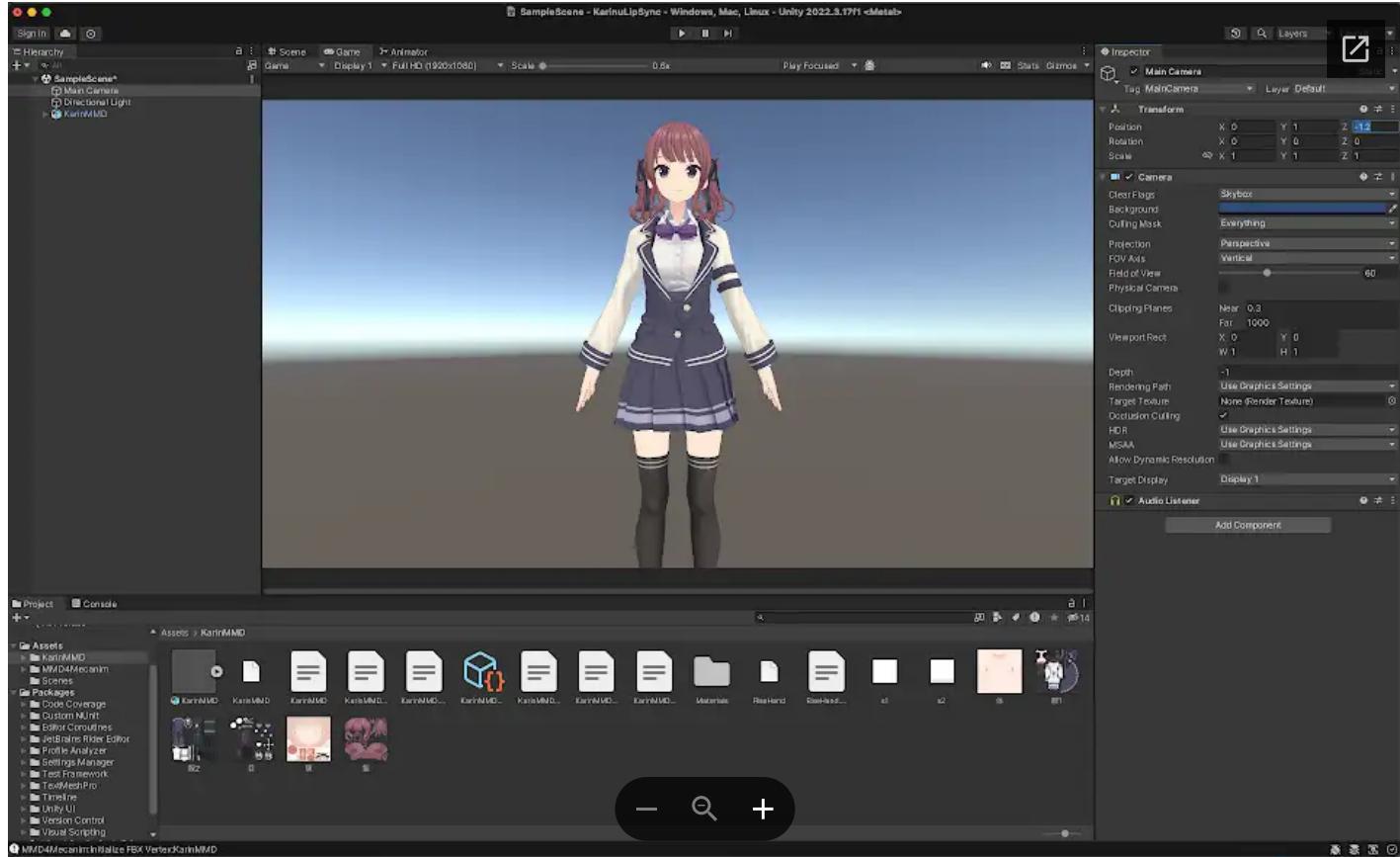
無事変換が終わったら**KarinMMD**フォルダ内の**KarinMMD.fbx**(右に矢印が出ているアイコン)を画面左側の**Hierarchy**タブ内の**SampleScene**にドラッグ&ドロップして配置してください。

無事に変換が成功していれば画面に夏色花梨3Dモデルが表示されるので拡大して確認してみます。

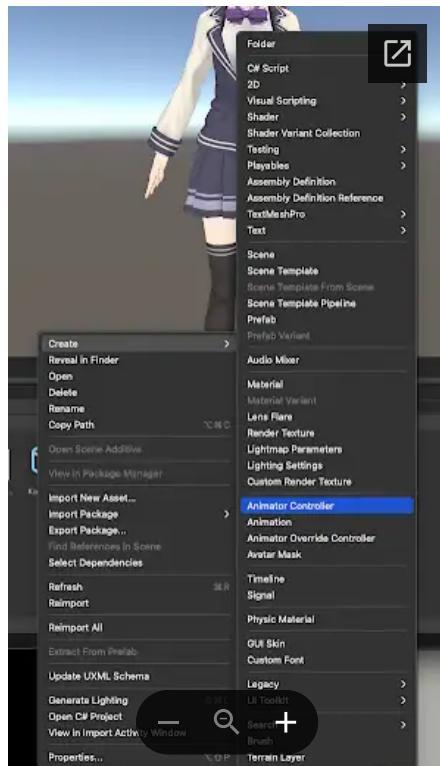


次に見やすいうようにカメラ位置を移動します。まず**SampleScene**内の**Main Camera**をクリックして右の**Inspector**から**Position**のZを-1.2に設定。次に同じく**SampleScene**内の**KarinMMD**を選択して右の**Inspector**から**Rotation**のYを-180に設定します。この-180度の回転はBlenderとUnityでそれぞれ座標系が違うためです。

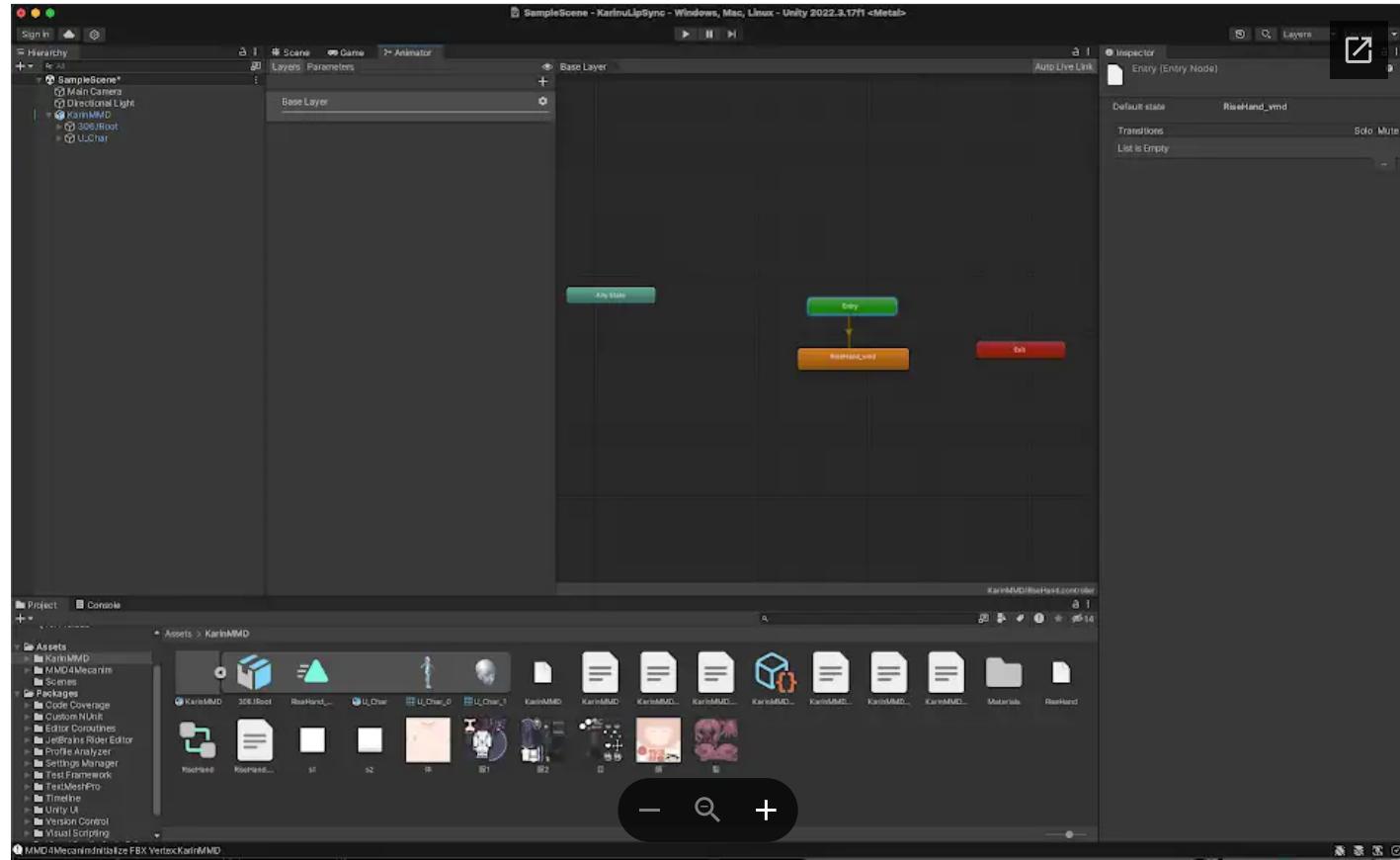
設定したら画面上部のタブを**Scene**から**Game**に切り替えてください。すると画面中央に夏色花梨3Dモデルが表示されると思います。



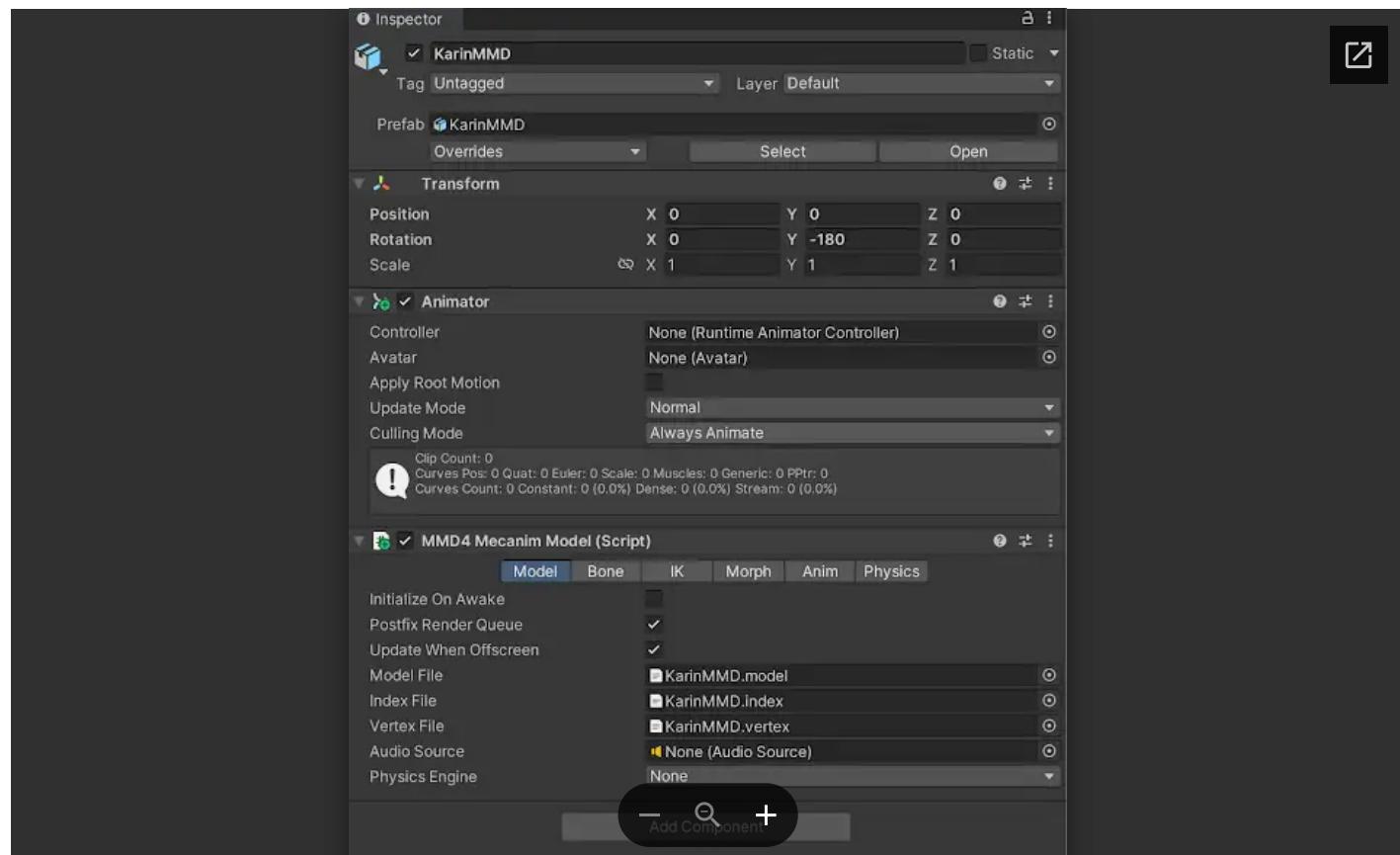
次にインポートしたアニメーションを確認します。場所はどこでもいいのですがとりあえず**KarinMMD**フォルダ内で右クリックをして**Create > Animator Controller**を作成します。名前は適当でいいですがわかりやすく**KarinAnimation**などの名前にしておくと良いと思います。



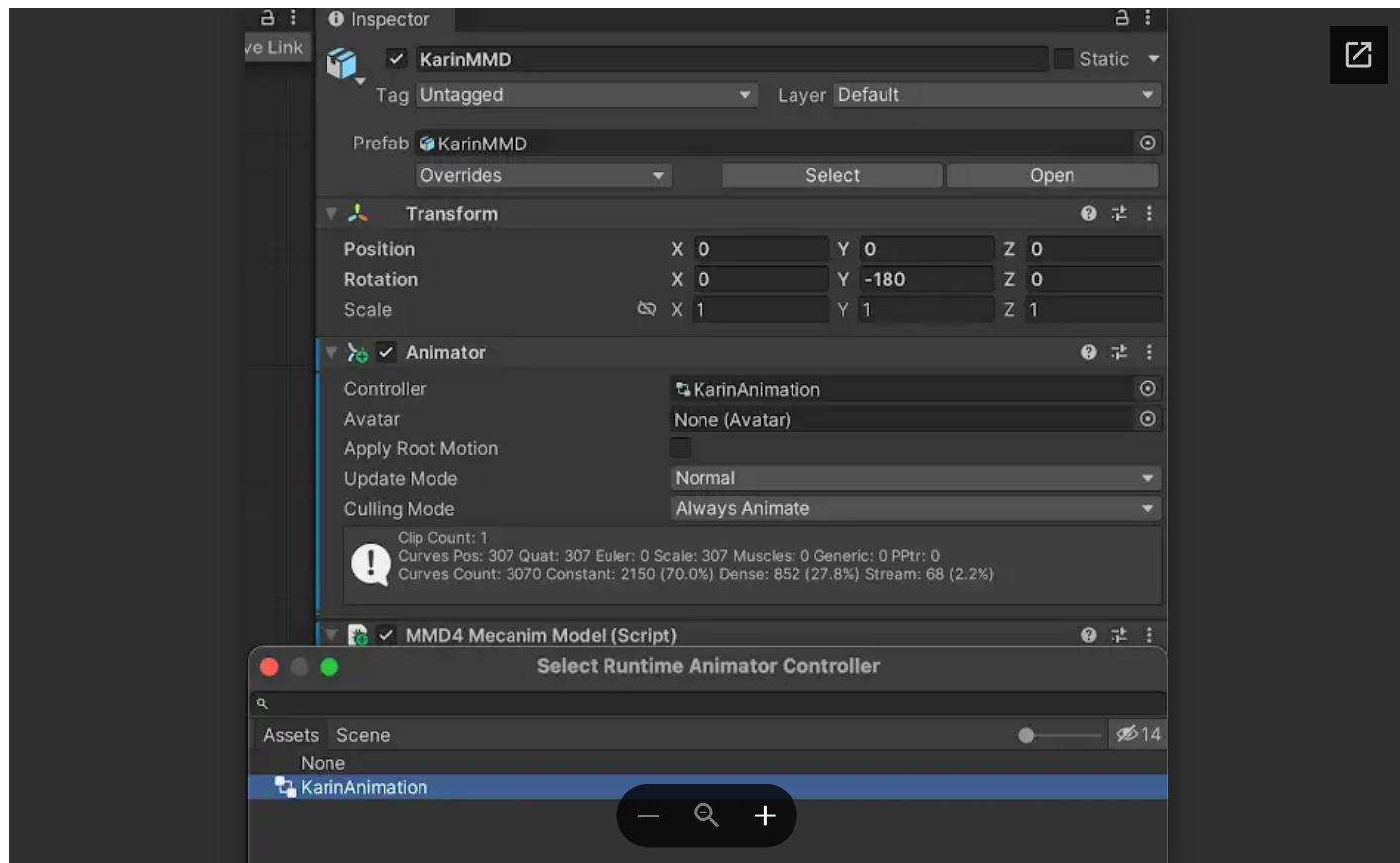
作成した**Animator Controller**をダブルクリックで開いたら**KarinMMD**フォルダの中にある右に矢印が出ているアイコンの**KarinMMD.fbx**の矢印をクリックし中身を展開します。その中の**RiseHand**がアニメーションコントロール用のファイルなので先ほど開いた**Animator Controller**にドラッグ&ドロップすると自動的に**Entry**と線でつながると思います。



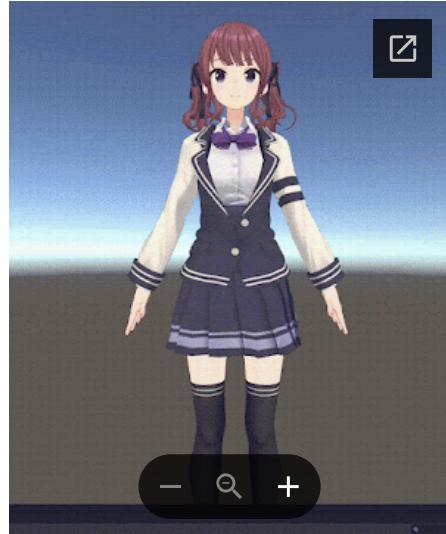
次に**SampleScene**内の**KarinMMD**を選択して右の**Inspector**から**Add Component**で**Animator**を追加します。自動的に**MMD 4 Mecanim Model (Script)**も追加されます。



追加したAnimatorのController欄から先ほどRiseHandを入れたAnimator ControllerのKarinAnimationを選択してください。



これでエディター上部の再生ボタンを押してBlenderで作った**RiseHand**のアニメーションが再現出来たらMMDモデルのUnityへのインポートはほぼ完了です。



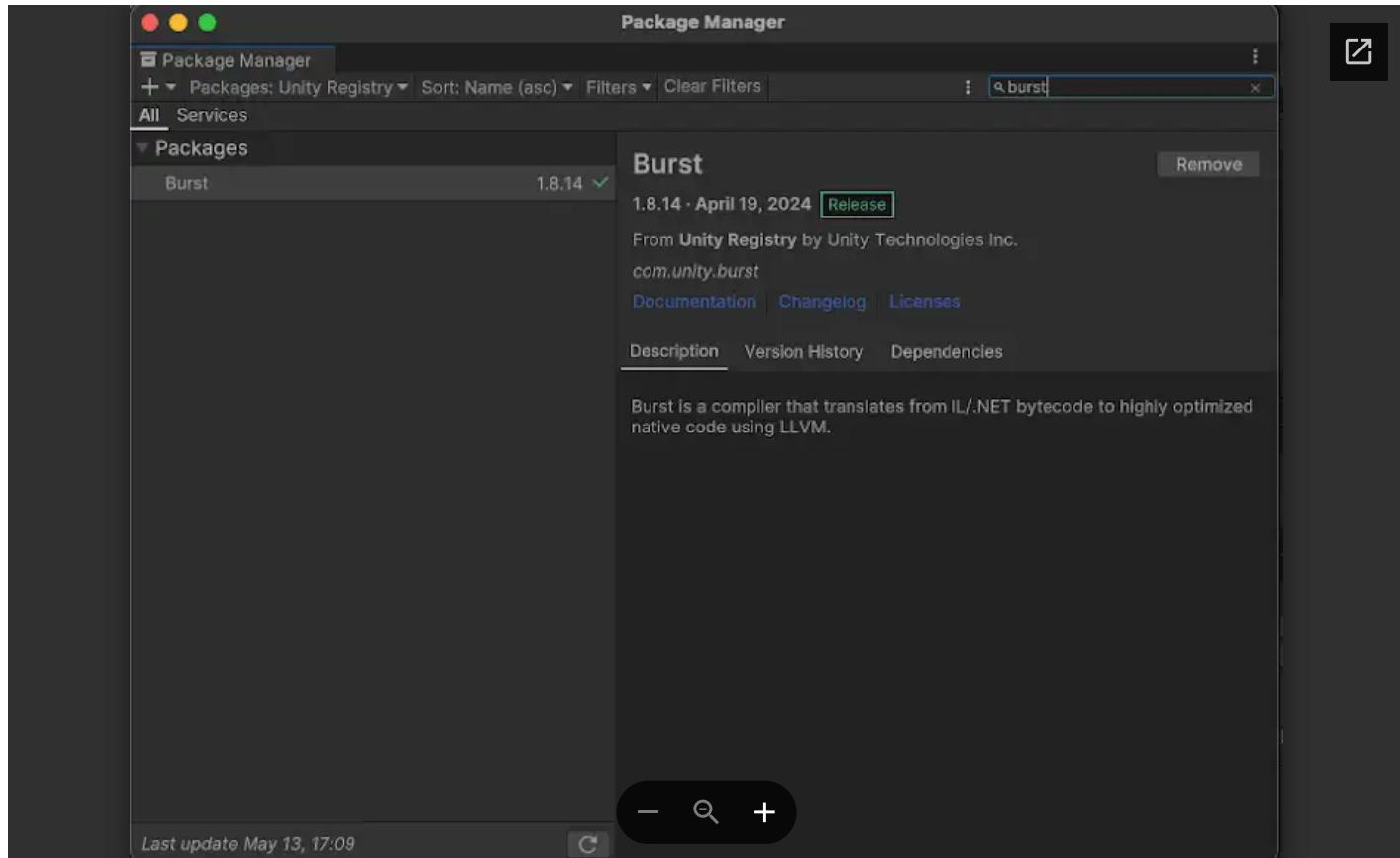
Unityのアニメーションの遷移はこの**Animator Controller**で各種アニメーションのパターンを用意して線でつなげたりスクリプトから制御するようになっています。

#リップシンク(口パク)させる

次に凹みさんのuLipSyncという3Dモデルにリップシンク(口パク)をつけるプラグインを導入します。

[uLipSyncのReleaseページ](#)から2024年5月時点での最新版v3.1.1 uLipSync-v3.1.1-with-samples.unitypackageをダウンロードして開いてください。
importしたら下の方で赤い文字でエラーが表示されていると思いますがこれはBurstというパッケージがインストールされていないためです。

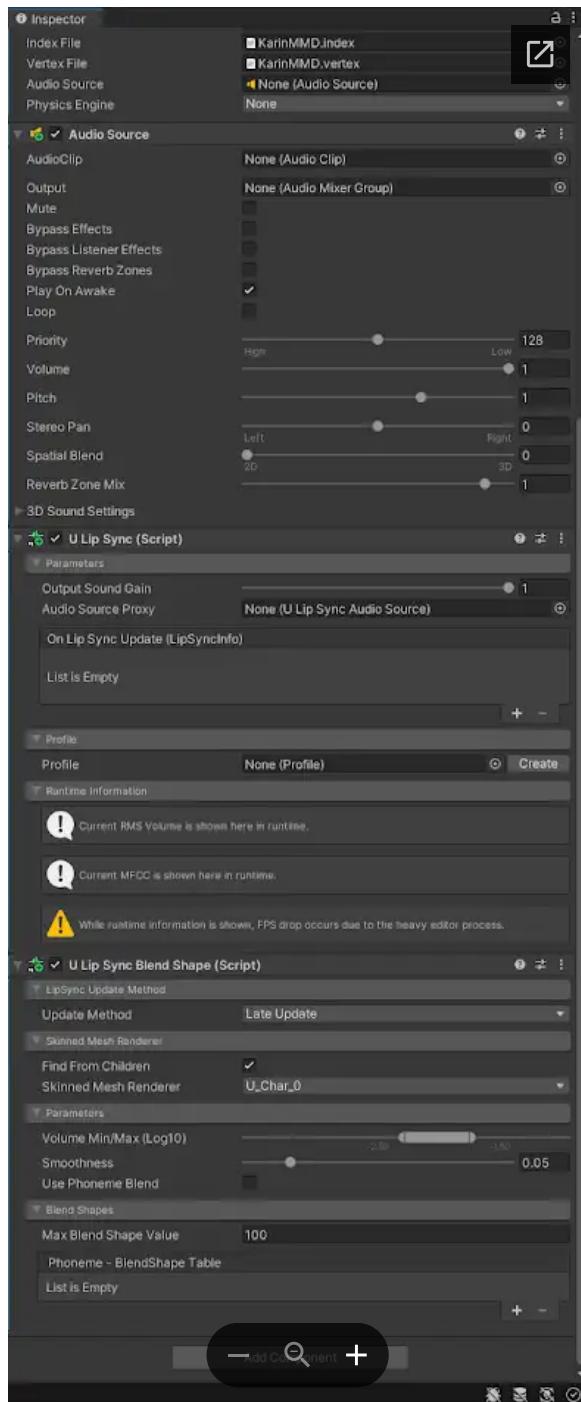
メニューのWindow > Package Managerを開き左上の**Packages: In Project**をクリックして**Unity Registry**に変更します。次に右上の虫眼鏡マークのサーチ欄にBurstといれ絞られた結果からBurstをインストールします。これでuLipSync導入の下準備が出来ました。



次にエディタ左側の**Hierarchy**タブの**Sample Scene**から**KarinMMD**を選択しエディタ右側の**Inspector**から**Add Component**を押しサーチ欄にと入力、絞られた結果中の**U Lip Sync(uLipSync)**を選択します。

もう一度**Add Component**を押して次は**U Lip Sync Blend Shape(uLipSync)**を選択しアタッチします。

最後にまた**Add Component**を押して次はサーチ欄にと入力**Audio Source**を選択しアタッチします。

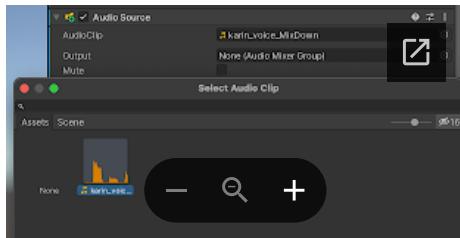


このように3つ**Audio Source**, **uLip Sync**, **ULip Sync Blend Shape** がアタッチされています。表示されている順番は特に関係ありません。

このアタッチしたうちの一つ**Audio Source**はオーディオファイルのプレーヤーです。この**KarinMMD**にアタッチしてある**Audio Source**に歌やトークの声のみが入ったオーディオファイルを読みませて、それをuLipSyncで解析して3Dモデルの表情を操作することで自動的にリップシンク(口パク)するという仕組みになっています。

口パク確認用として[Synthesizer V](#)で夏色花梨の「あいうえお」を出力したものをおきます。これをダウンロードしたら**KarinMMD**フォルダにでもドラッグ&ドロップしてください。

先ほどuLipSyncといっしょにアタッチした **AudioSource**の **AudioClip**欄から先ほどダウンロードした「あいうえお」の**karin_voice_MixDown.wav**を指定します。

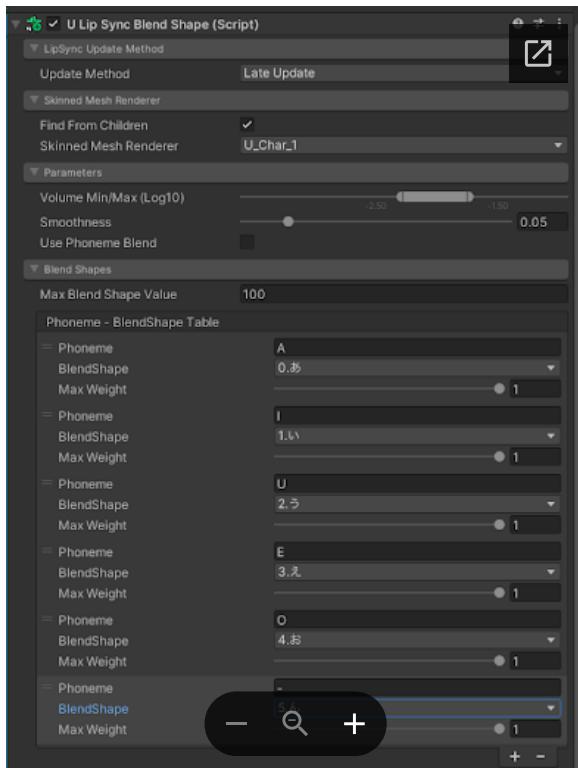


次に**U Lip Sync Blend Shape**の設定を下の画像を画像を参考にしながら設定してください。具体的な変更点としては**Skinned Mesh Renderer**がデフォルトでは**U_Char_0**なのを**U_Char_1**に変更します。

これは3Dモデルもいくつかのパーツに分かれていたりして統一されてはいないのですが口を含んだパーツを指定してあげていると考えると良いと思います。なので違うモデルだとU_Char_XXのXXは変わると思ってください。

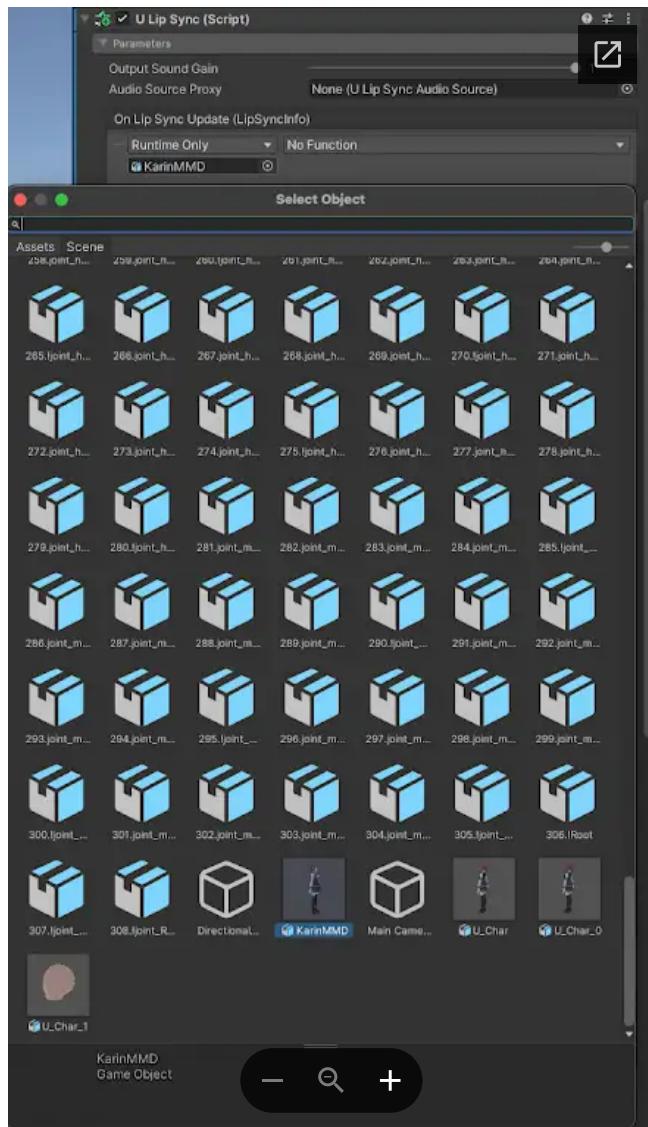
もう一つの変更点はBlend ShapesのPhoneme - BlendShape Tableです。

*Phoneme(音素)*とMMDの表情などをコントロールするBlendShapeを指定してあげます。要はuLipSyncが検出したA I U E O -(無音)とMMDの「あいうえおん(無音)」の口の変化をそれぞれ対応させてる箇所です。

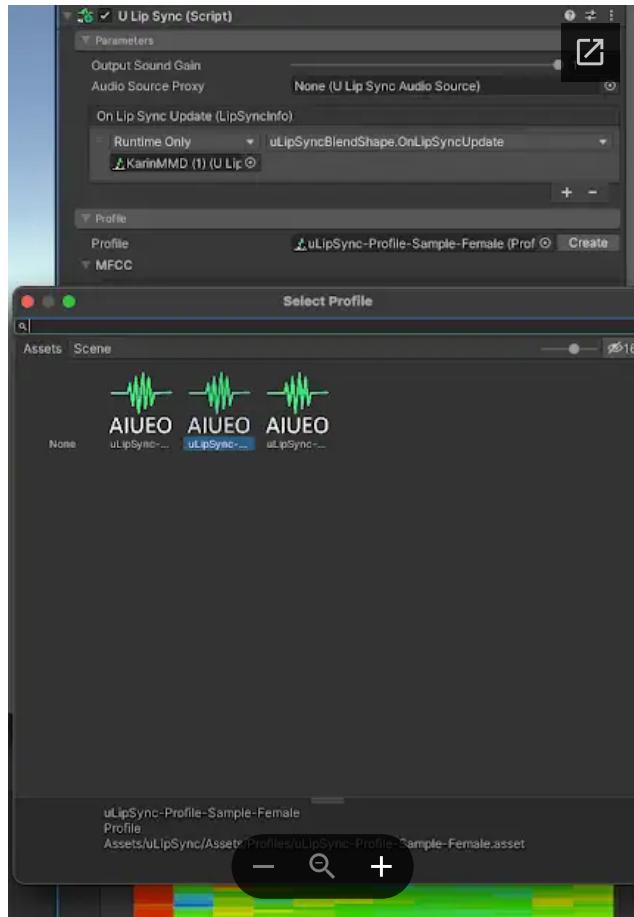


次にU Lip Syncの設定をします。

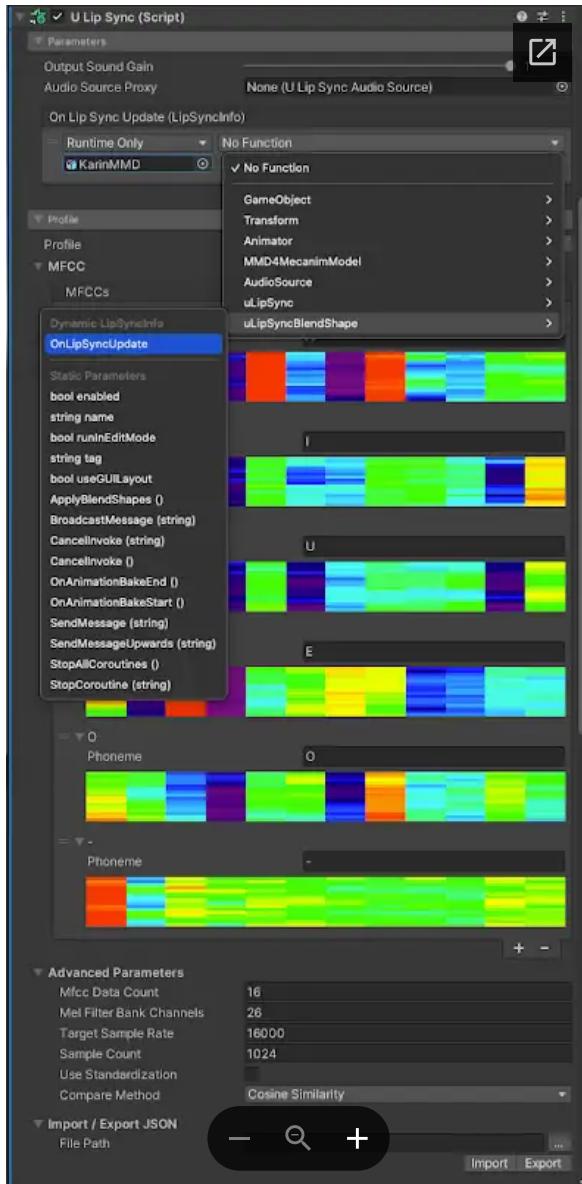
U Lip SyncのOn Lip Sync Update(LipSyncInfo)のList is Emptyとなっている箇所の右側の+ボタンを押し、現れたNone** (Object)となっている箇所をクリックAssetsとSceneというタブのSceneを選択してサーチ欄にKarinと入力して絞られたKarinMMDを選択します。



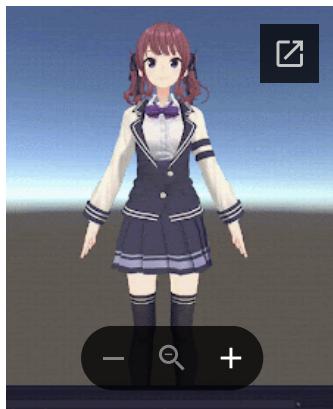
次に**Runtime Only**の右側**No Function**になっている箇所をクリックし**uLipSyncBlendShape > OnLipSyncUpdate**を選択します。



もう一箇所Profile欄を選択してLipSync-Profile-Sample-Femaleを選択しておきます。これは音声を解析する時の元データのようなもので、キャリブレーションして自作することも出来るのですがとりあえずサンプルのものを利用します。



これでuLipSyncの設定が終わったので画面上部の再生ボタンを押してみて「あいうえお」とリップシンクしてくれたらOKです。uLipSyncはキャリブレーションといって声の特徴を解析してさらに検出の精度を上げる方法もあるので詳しくは[uLipSync](#)を参考にしてみてください。

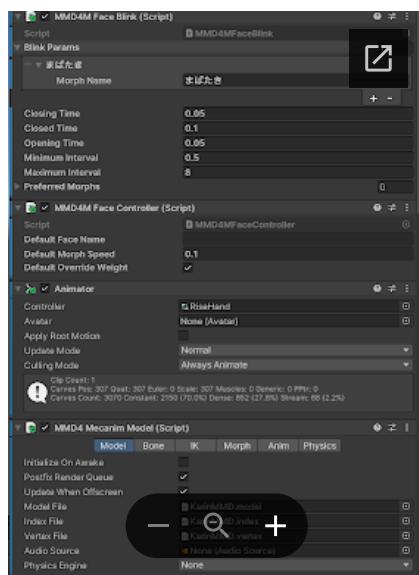


#まばたきさせる

これでやりたい事出来た感がある感じですけど、もう一つやっぱりまばたきは生理的なものなのでないとななんか不自然な感じがします。そこで[udasan](#)の[MMD4MFace](#)を利用して自動でまばたきさせてみます。[MMD4MFaceのページ](#)に行き右上にある**Code**というとこをクリックして[Download Zip](#)でZipファイルをダウンロードしたら解凍して[KarinMMD](#)フォルダと同じようにUnityエディタのAssetsに解凍したフォルダごとドラッグ&ドロップしてください。

次にエディタ左側の**Hierarchy**タブの**SampeScene**から[KarinMMD](#)を選択しエディタ右側の**Inspector**欄に先ほどAssetsに入れた[MMD4MFace](#)のフォルダに入っている[MMD4MFaceController.cs](#)と[MMD4MFaceBlink.cs](#)をそれぞれドラッグ&ドロップしてアタッチしてあげます。

アタッチしたらMMD4M Face Blink(Script)のBlink ParamsのMorph Nameにまばたきと入れてください。これでまばたきの設定は終わりです。
再生ボタンを押してみると自動的にまばたきするようになってると思います。



#これを土台にしたMVの作り方

これで基本的なMMDのモデルの導入とリップシンクが出来ました。

MVを作る時はもう一つ **AudioSource**を用意してそこにボーカル以外の音が入ったトラック(要はカラオケ)で、uLipSyncがアタッチされている **AudioSource**には逆にボーカルのみが入ったトラックをそれぞれDAWで曲の頭から終わりまで同じ長さで書き出してあげて読ませるとUnity内で同時再生されバーチャルカラオケ的な事になります。

それを**Unity**の**Recorder**で画面を録画して、それを動画編集ソフトなどで音源をマスタリングしたものに差し替えてあげて歌詞のテキストをつけたりしてMVにするというような方法で作ってます。

Unityはゲームを作成するソフトなのでいろいろなエフェクトだったりもあるし、それを例えば同じように **AudioSource**に読ませたトラックの音量を検出して花火を打ち上げたりとか演出に使う事が出来るので興味がある人はやってみると楽しそうです。